



IoT based Smart Parking System

Muhammad Mehran (Registration # 979)

Muhammad Husnain (Registration # 974)

Ehsan Khalid (Registration # 995)

Supervised by

Mr. Shahid Mahmood

Riphah College of Computing

Riphah International University

Faisalabad, Pakistan

Intellectual Property Right Declaration

This is to declare that the work under the:

IoT based smart parking system carried out in partial fulfilment of the requirement of:

☐ **BS (CS) FYP** ☐ **BS (SE) FYP** ☐ **MS Project/Thesis** ☐ **Course Project**

Is the sole property of the Riphah International University, and is protected under the intellectual property right laws and conventions. It can only be considered/used for purposes like extension for further enhancement, product development, adoption for commercial/organization usage, etc. with the permission of the University.

This above statement applies to all students and faculty.

Date: _____

Student 1

Name: Muhammad Mehran

Signature: _____

Student 2

Name: Muhammad Husnain

Signature: _____

Student 3

Name: Ehsan Khalid

Signature: _____

Supervisor

Name: Mr. Shahid Mahmood

Signature: _____

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the:

Title: IoT based smart parking system

Is the sole contribution of the author(s) and no part here of has been reproduced on as it is basis (cut and paste) which can be considered as **Plagiarism**. All referenced parts have been used to agree the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined

Date: _____

Student 1

Name: Muhammad Mehran

Signature: _____

Student 2

Name: Muhammad Husnain

Signature: _____

Student 3

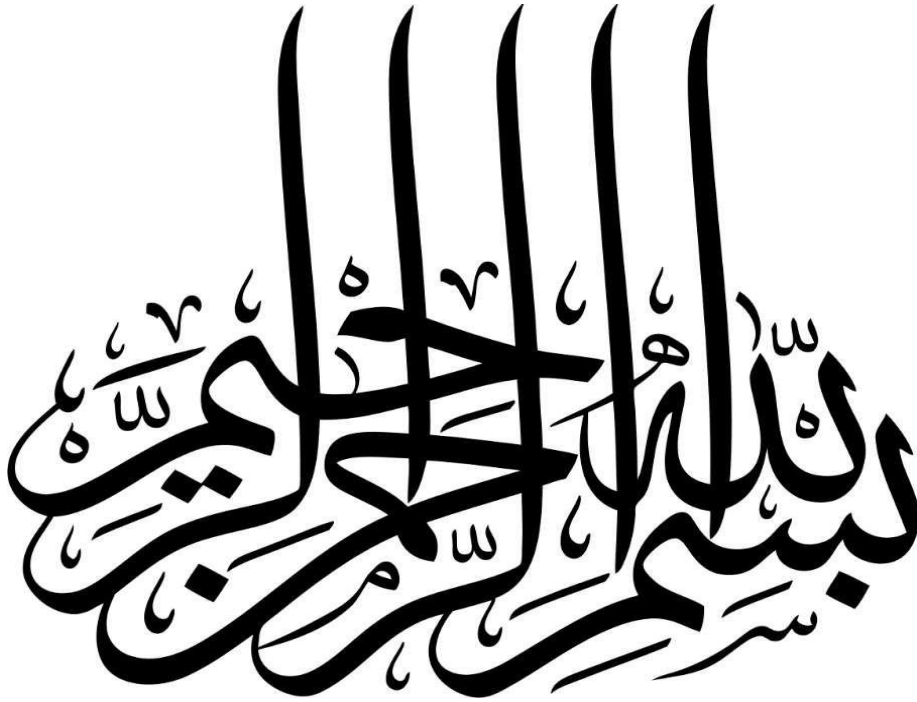
Name: Ehsan Khalid

Signature: _____

Supervisor

Name: Mr. Shahid Mahmood

Signature: _____



IN THE NAME OF ALLAH, THE MOST BENEFICIENT THE MOST MERCIFUL

Dedication

I dedicate this project to my respected parents, teachers, siblings, and friends whose care, love and guidance have been nurturing me since I was born. They have been my source of inspiration, and my beacon of guidance.

Abstract

Searching a parking space has become a daily anxiety these days, and that is where the inspiration for this idea came up from. **“Internet of Things based smart parking system”** is unique available solutions to at least lessen the traffic congestion produced due to finding parking spots. This system will provide information about the occupancy status of the spaces in the parking spot equipped by devices that notice the existence of vehicles. **“Smart parking android application”** will be connected to system with real time database and will show real time data received from system to user and will help drivers to make cool decisions which will lessen congestion.

Acknowledgements

We would like to express our deepest appreciation to all those who provided us the possibility to complete our final year project. A special gratitude we give to our final year project supervisor, Mr. Shahid Mahmood, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project.

Furthermore, we would also like to acknowledge with much appreciation the crucial role of the staff of department of computing, who gave the permission to use all required equipment and the necessary material to complete this project. I have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advices.

Table of contents

Abstract.....	I
Acknowledgements.....	II
Document conventions.....	A
Chapter # 1	1
1 Introduction	2
1.1 Development procedure.....	3
1.1.1 Requirements analysis	3
1.1.2 Specification	3
1.1.3 Software architecture	3
1.1.4 Implementation	3
1.2 Process model we are using.....	4
1.2.1 Why we choose Iterative model for our project?.....	5
Chapter # 2.....	6
2 Literature review	7
2.1 Technologies used in smart parking system.....	7
2.1.1 Smart Parking System Based on Fuzzy logic	7
2.1.2 Smart Parking System Based on (V2I)	8
2.1.3 Smart Parking System Based on Global Positioning System GPS.....	9
2.1.4 Smart Parking System Based on Computer Vision	9
2.1.5 Smart Parking System Based on RFIF Technology	10
2.1.6 Challenges of the smart parking Systems	11
2.2 Sensors used in existing system.....	11
2.2.1 Active infrared sensor	11
2.2.2 passive infrared sensor	12
2.2.3 Piezoelectric sensor.....	13
2.2.4 Ultrasonic sensor.....	13
Chapter # 3.....	15

3	Project vision.....	16
3.1	Problem statement	16
3.2	Problem background.....	16
3.3	Proposed solution	16
3.3.1	Software	17
3.3.2	Hardware.....	17
3.4	Project objectives.....	17
3.4.1	Business opportunity.....	17
3.4.2	Improved parking.....	17
3.4.3	Reduced traffic.....	17
3.4.4	Reduced pollution	18
3.4.5	Decreased Management Costs	18
3.5	Project scope.....	18
3.6	Constraints	18
Chapter # 4.....		19
4	Software requirement specification.....	20
4.1	List of features	20
4.1.1	Registration	20
4.1.2	Login	20
4.1.3	Find Parking space.....	20
4.1.4	Select parking space.....	20
4.1.5	Book a parking space	20
4.1.6	Make payment.....	20
4.1.7	Users verification and authentication.....	21
4.1.8	Navigation.....	21
4.1.9	Notification	21
4.1.10	Received bookings	21
4.1.11	Check active session	21
4.1.12	Rent out parking space.....	21
4.1.13	Account settings.....	22
4.1.14	Show parking details.....	22

4.2	Functional requirements	22
4.2.1	Services	22
4.3	Non-Functional Requirements.....	22
4.3.1	Responsiveness	22
4.3.2	Efficiency.....	23
4.3.3	Maintainability.....	23
4.3.4	Modularity.....	23
4.3.5	Usability.....	23
4.3.6	Easy to Use	23
4.3.7	Space Requirement	24
4.3.8	User Interface Specifications	24
4.3.9	Reliability.....	24
4.3.10	Understandability	24
4.3.11	Simple program.....	24
Chapter # 5.....		25
5	High level use cases	26
5.1	Use case diagram.....	26
5.1.1	System Use cases for Smart parking based on internet of things	26
5.1.2	Use cases for unauthorized user.....	27
Chapter # 6.....		28
6	System detailed design	29
6.1	Use case of authorized user	29
6.2	Flowchart.....	30
6.2.1	Overall Spark application flow chart	31
6.2.2	Overall Spark hardware flow chart	32
6.3	Sequence diagram.....	33
6.3.1	Sequence diagram of spark system	34
6.4	Hardware prototype diagrams	35
6.5	Hardware circuit diagrams.....	36
Chapter # 7.....		37

7	Implementation.....	38
7.1	Software.....	38
7.1.1	Software We Use	38
7.1.2	Build developing environment of Android	39
7.1.3	Installation steps of the developing environment	39
7.1.4	Why we use Android Platform?	39
7.1.5	Firebase	39
7.2	Hardware We use.....	41
7.2.1	NodeMCU Microcontroller	41
7.2.2	HC-SR04 ultrasonic sensor.....	42
7.2.3	Servomotor.....	43
7.2.4	LCD for display	43
7.2.5	RFID System.....	44
7.2.6	Wi-Fi Module.....	44
7.2.7	I2C LCD.....	45
7.3	Front end designing using XML.....	45
7.3.1	Components of front end designing	46
7.4	Back end development using JAVA.....	47
7.4.1	Components of backend development	48
Chapter # 8.....		52
8	Testing and results.....	53
Chapter # 9.....		55
9	Conclusions and future work	56
Chapter # 10.....		57
10	User manual	58
10.1	Software	58
10.1.1	Starting application	58
10.1.2	Create account with Email	59
10.1.3	Permissions	63
10.1.4	Application navigation drawer.....	66

10.1.5	Profile settings	67
10.1.6	Account verifications	71
10.1.7	Search parking space.....	72
10.1.8	Reservation	73
10.1.9	Active session	77
10.1.10	Notifications.....	78
10.1.11	Extend parking time.....	79
10.1.12	Add parking space.....	82
10.2	Hardware	83
Chapter # 11	84
11	bibliography	85
11.1	Appendix A list of figures.....	85
11.2	Appendix B list of tables.....	87
Chapter # 12	89
12	References.....	90

Document Conventions

Paper	Standard A4 Size. Width: 8.27” Height: 11.69” Weight: 90 Grams
Fonts, Type Styles	Title = 24 bold (Times New Roman) Sub-Title = 18 bold (Times New Roman) Heading I = 16 bold (Times New Roman) Heading II = 14 bold (Times New Roman) Font Size = 12 (Normal Text) (Times New Roman) Line Spacing = 1.5 Paragraph Spacing = 6 pts
Margins	Top = 1” Bottom = 1” Left = 2” Right = 1” Margin = Justify (Normal Text)
Headers/Footers	The header will comprise the title of the Project report. On every odd page will appear the title of the report while on the even pages the title of the chapter or section will be mentioned. The first page of every section or chapter shall not carry the header. Footer will comprise the name of institute and page number. Footer center will have name of university and footer right contain page number of report.
Spark	Android application name stands for “smart parking”
Spark System	Hardware system stands for “smart parking system”
IoT	Internet of Things
IoT based Smart parking system	Complete project name which includes android application “Spark” and hardware system “Spark System”.

Table 1 Table of conventions

Chapter # 1

Introduction

1 Introduction

Finding an empty parking space during peak hours in areas like Universities, Shopping malls, Hotels, Airports, Hospitals and Convention Center has been always irritating for every vehicle owner. Surveys tell that traffic produced by cars finding for positions in Parking Spaces is up to 40% of the entire traffic. Searching a parking space has become a daily anxiety these days, and that is where the inspiration for this idea came up from.

Present days that is a serious problem to look after, and “**IoT based Smart parking system**” is one off unique available solutions to at least lessen the traffic congestion produced due to finding parking Spots. The idea of "IoT" is network of devices, home applications, and different things inserted with hardware, programming, sensors and system availability which enable these items to interface and trade information.

Through the advancement of tools, sensors that sense the existence of anything and our idea is having a system where parking spaces are equipped with sensors that states exactly the occupancy status of the parking places and a central controlling system that posts this occupancy status to real time database.

The proposed System will use ultrasonic sensor to sense the existence of a vehicle (whether the parking slot is free or not). Based on the parking slot occupancy, the status (occupied/unoccupied) is showed on the mobile application. In real time, the setting has sensors and devices fixed into parking places, transferring data on the occupancy status and the vehicle drivers can find for parking availability via their mobile phones. Therefore, the driver would identify where there is a vacant spot to park his vehicle in fewer time, decreasing the energy intake and air pollution.

When user find free parking space with proposed mobile application he/she will also have option to book this parking space for a time period, system will also have its own payment method to receive and send payments from users to space owner.

Everyone who has some free space in his/her house or organization and he/she wants to earn some money from it he/she can add their space in proposed system.

This parking space will be shown on user mobile application with information that space owner provided. Using mobile application driver can contact with space owner to confirm booking.

1.1 Development procedure

Here are some details about development procedure that will be followed to develop a quality product or system.

1.1.1 Requirements analysis

Separating the requirements of a coveted software artefact is the primary task in developing it. While clients likely trust they comprehend what the software is to do, it might need expertise and knowledge in software engineering to perceive inadequate, uncertain or opposing requirements.

1.1.2 Specification

Specification is the undertaking of absolutely depicting the software to be developed, in a numerically thorough manner. Actually, best details are composed to comprehend applications that were previously created, while security-critical software are often cautiously specified rather than application development. Specifications are most critical for outside interfaces that must stay stable.

1.1.3 Software architecture

The architecture of a software alludes to a dynamic representation of that system. Architecture is worried about ensuring the software system will meet the prerequisites of the artefact, and in addition guaranteeing that future requirements can be tended to.

1.1.4 Implementation

Proposed “**IoT based smart parking system**” is a based on NodeMCU and mobile application that is established to help drivers in searching available parking space during any time 24/7. The main goal of the System is to provide the Real Time occupancy status (empty/unavailable) of the parking spaces. The authorized user is permitted to select any date and can get the information related to parking spaces occupancy during that day. We divided our system into

two parts, 1st part which is a software based mobile application and 2nd part is based on hardware.

The authorized user is permitted to select any time and can get the information related to parking spaces occupancy during that day.

We divided our system into two parts, 1st part which is a software based mobile application and 2nd part is based on hardware.

1.1.4.1 Spark

It is a mobile application which shows that has Real Time occupancy statuses of the parking slots. The application is refreshed when any changing occurs in parking space and the latest information is displayed for the user. This mobile application will be developed by using android software development kit tools (SDK). Application will use Maps application programming interface (API) provided by google to show the current status of parking spaces and to navigate the user to his destination or spot where user parked vehicle. The system will also use firebase real time database to send and receive data.

1.1.4.2 Spark system

The Physical part of System which is consist on NodeMCU board and ultrasonic sensors to detect the presence of objects in parking spaces it sends and receives Real Time data on firebase database. We use RFID tag and antenna in this system to recognize authorized user. System checks that user is Authorized or not, if user is unauthorized than parking gate will not open.

1.2 Process model we are using

A project can be divided into many phases or different tasks, sub-tasks and the activities that to be needed to be completed in specified timeline. So, to maintain the core aspects of project. i.e. time resources and cost every project should follow a Software Development Life Cycle. As far as ours project is concerned we choose “Iterative Model” as the development cycle.[1]

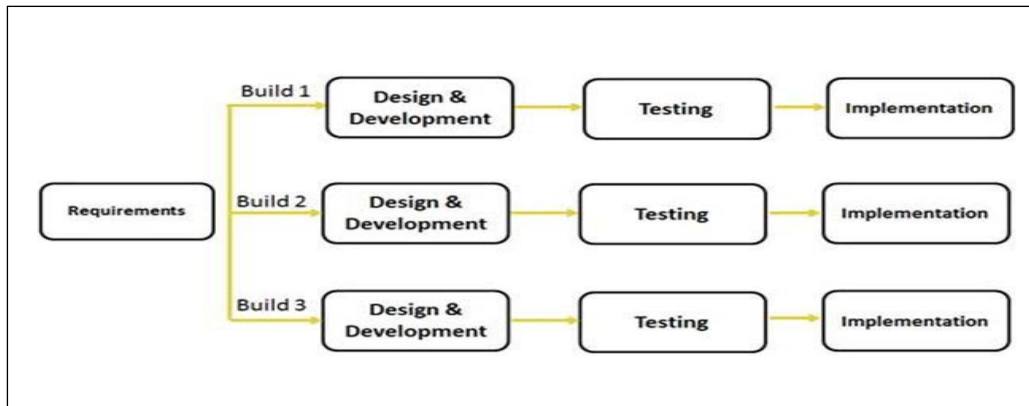


Figure 1-1 SDLC - Iterative Model

1.2.1 Why we choose Iterative model for our project?

An iterative life cycle model does not endeavor to begin with a full determination of requirements. Rather, improvement starts by indicating and actualizing simply part of the product, which would then be able to be assessed with a specific end goal to distinguish advance requirements.

This procedure is then rehashed, delivering another variant of the product for each cycle of the model. In the diagram, when we work iteratively we make unpleasant item or item piece in one iteration, at that point audit it and enhance it in next iteration and so on until the point that it's done.

Chapter # 2

Literature Review

2 Literature Review

The smart parking systems were originally applied in Europe, Japan and USA but later on as the other countries started developing, these smart parking systems are being installed in these countries as well. With more improvement in the smart parking systems the problem of finding empty spaces and all the trouble is going to remove.

This unit discusses the different methods used for smart parking. Also in this section, all related studies are collected into groups based on the techniques used (conference papers, articles) specifically in the academic field. It is very clear from all the references below that categories and classifications of smart parking vary from source to source. Some rely on the technology used while others rely on data processing to get information about the parking statues. For example, in the centralized supported parking search, the information processing will be stored on the central processor (server). The non-assisted parking search does not have a server, and no information will be provided to a user. Numerous technological methods are grouped into the following classification.

2.1 Technologies used in smart parking system

2.1.1 Smart Parking System Based on Fuzzy logic

Fuzzy logic is a methodology to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based.

The idea of fuzzy logic was first advanced by Dr. Lotfi Zadeh of the University of California at Berkeley in the 1960s. Dr. Zadeh was working on the problem of computer understanding of natural language. Natural language (like most other activities in life and indeed the universe) is not easily translated into the absolute terms of 0 and 1. (Whether everything is ultimately describable in binary terms is a philosophical question worth pursuing, but in practice much data we might want to feed a computer is in some state in between and so, frequently, are the results of computing.) [2]

Proposed a system that depends on an FPGA-based fuzzy logic controller (FLC). The benefit of using a field-programmable gate array (FPGA)-based FLC compared to software FLC is

that takes less time to process the information. First, a Fuzzy Control System is chosen. Then, the implementation of the fuzzy rule-based system takes place upon the neural network architecture.

It is the main reason for learning and adapting from the training data: “The neuro-fuzzy system has the ability to reason like human beings as well as it has expert knowledge”.

Proposed that an RF transceiver and antenna with an ATmega 128 L micro-controller system could operate by monitoring the availability of car-parking spaces and send this information to customers and facility administrators. presented a fuzzy approach for the control of the backward movement of trucks and trailers in a dynamic environment. This method was then expanded to circumstances in which there are obstacles in the truck’s pathway. In the first scenario, it is assumed the obstacles are constant. The second scenario assumed by the authors is that there are moving obstacles which can mean the truck must be directed to the parking facility. The parking process is completed due to the intelligence of fuzzy logic.[3]

2.1.2 Smart Parking System Based on (V2I)

System using the term (CVT) to refer to Connected Vehicle Technology which depends on wireless data transmission between vehicle and infrastructure (V2I). This promising technology emerged recently. It proposes a new smart parking technique that depends on developing a new VANET-based smart parking to be used for smart parking. It refers to Vehicular Communication Systems, in which vehicles and roadside units are the communicating nodes, that is, they communicate and exchange information with each other, such as safety warnings or supplying the traffic congestion information and even for finding vacant parking spaces. Basically, vehicular networks are considered to contain two types of nodes: vehicles and roadside stations Both are categorized under the term ‘Dedicated Short Range Communications’ (DSRC) devices. DSRC works in 5.9 GHz bands with a bandwidth of 75 MHz and a range of about 1000 m. This is a two-way method of communication including Vehicle-to-Infrastructure (V2I) and Infrastructure-to-Vehicle (I2V) communication. In the “smart parking” system, usually, V2I communication includes drivers sending their parking requests, providing driver information, and confirming that reservation to the system.[4]

2.1.3 Smart Parking System Based on Global Positioning System GPS

Global Positioning Systems (GPS) technology is used to determine and track a vehicle's precise location.

In this domain, it is used to offer information about the location and availability of parking spaces at the destination. This location-based system called NAPA. The server in the system associates buildings on the campus with parking lots in the order of distances to the building. After locating the nearest available parking lot, the user sends the NAPA server a message that he/she has parked. Then the server updates the information about the lot accordingly.

When the user leaves the parking lot, the NAPA server can automatically charge the appropriate parking fee if necessary. This new smart parking system uses SMS service. This system is capable of finding parking spaces in specific car park areas.

A parking reservation system is developed in such a way that users can book their parking spots over short message services (SMS) using the GPS. The SMS is processed by a wireless communication instrumentation device called a micro-RTU (Remote Terminal Unit).[5]

2.1.4 Smart Parking System Based on Computer Vision

This field of study includes methods for acquiring, processing, and analyzing images. It uses computers to emulate human vision, including learning and being able to make inferences and take actions based on visual inputs, also called computer vision. The goal of computer vision is to make computers efficiently perceive and process visual data, such as images and videos, and act upon changes in these images. Usually, the technique involves analyzing a few frames per second and then sending the data to a central database, after which, the user can retrieve information about the changes at the parking lot. This system utilized CCTV in a vehicle detection stream to detect the presence of a car or vehicle in a particular parking lot. Pixel detection is used to detect the presence of a vehicle in each parking lot. A certain number of pixels in the grayscale are used as the threshold to differentiate pixels from the vehicle and from the unoccupied lot. Another parking system, called CCTV, uses images to detect parking spaces. CCTV cameras are fitted in car parks to automatically detect car parking spaces.

However, these methods may incorrectly detect parked vehicles. The system is targeted on cases where occupancy values are required. This system uses CCTV cameras that are fitted in car-parks to automatically detect car parking spaces. However, these methods are not always accurate in cases where occupancy values are required.

Motivation for developing this system came from the fact that minimum cost is involved because image processing techniques are used rather than sensor-based techniques. This project is called the Car-Park Occupancy Information System (COINS).[6]

2.1.5 Smart Parking System Based on RFIF Technology

In many academic papers, smart parking's RFID solutions make it possible to manage permit parking easily, especially in the prototype stages. The main mechanism of RFID technology depends on an electromagnetic field to identify and track tags attached to objects automatically. The system uses a software program for controlling and reporting changes in the status of the parking space, and for the operation of tasks such as choosing the closest vacant parking space, and it then sends the information to the driver. Meanwhile, in this system containing Gate-PC Controller and Embedded Gate Hardware, an RFID System, and a Modular Parking Management Platform: "Most systems in the Modular RFID Parking Management System are modulated and can be substituted for any other similar system or hardware".[7]

Technique Based	Reliability	Communication method	Circuit Complexity	Detection Accuracy
RFID	High	Wi-Fi	Complex	Accurate
CCTV coins	High	Wi-Fi a, g	N/A	False detection may occur
light sensor	High	Zig-bee	Complex	Accurate at day time Cannot be used at night
Acoustic sensor	High	RF	Complex	Seriously influenced by environmental noise
Optical sensor	High	Blue-tooth	Complex	Very accurate
Ultra-sound	High	Switch and LAN	Simple	Accurate
SMS	High	GPS	Simple	Accurate
Magnetic sensors	High	WIFI /RF	Simple	Accurate
Infrared	High	RF/Wi-Fi	Simple	Too sensitive Maximum accurate at day time

Table 2 Overview of smart parking based on different techniques

2.1.6 Challenges of the smart parking Systems

There are major challenges facing today's transportation systems and drivers on a daily basis regarding special parking systems for which smart city engineers and designers have to be prepared. Numerous recent studies have led to the conclusion that new smart parking systems are needed in almost every metropolitan city in the world especially in the next ten years to alleviate many problems, such as petrol consumption and pollution emission, and to improve time-saving and reduce frustration when looking for a parking space. Therefore, for any proposed system to be considered smart in relation to the parking process, it should have as a minimum, the following factors and specifications:

- Be able to accurately sense vehicle occupancy in real-time
- Provide guidance for users about available parking
- Simplify the parking experience and add value for parking stakeholders, such as drivers
- Enable intelligent decisions to be made using data, including real-time status applications, and historical analytic reports
- Be able to provide the user with all the necessary information about the status of any changes in the parking area that might happen in real time

These challenges must be addressed from the very beginning to ensure that the system will work efficiently. Many studies related to traditional smart parking systems in the last.

2.2 Sensors used in existing system

2.2.1 Active infrared sensor

Active Infrared sensors are configured to detect the presence of a vehicle by using infrared energy emitted towards the vehicle and then measuring the energy that has been reflected of it. It can be used in different conditions and operations.

The sensor is pretty accurate in determining the vehicle position but is a bit sensitive to the outdoor environment conditions.

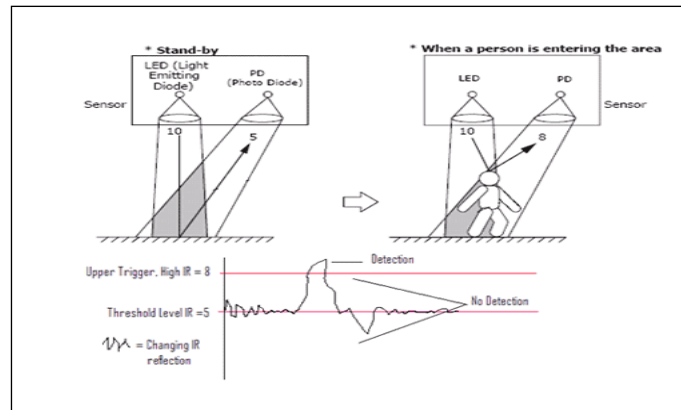


Figure 2-1 Active Infrared Sensor

2.2.2 passive infrared sensor

The passive infrared sensors operate on the pyroelectric technology which detects the thermal infrared emissions from the vehicles and thereby converting the signal received into a DC output. Since they measure the infrared radiation instead of emitting the same, they are known as Passive Infrared Sensors. This type of system is generally mounted above ground for a wider field of view. When a vehicle gets into the vicinity of this field, a significant change is caused in the signal received by the sensor and compared to the steady state when there isn't any vehicle. The vehicle speed and length can also be

inferred using the above technology, hence making it easier to identify the type of vehicle (2-wheeler, 4-wheeler etc.).

These systems are sensitive towards the weather as their performance depreciates in conditions like snow falls, heavy rains etc.

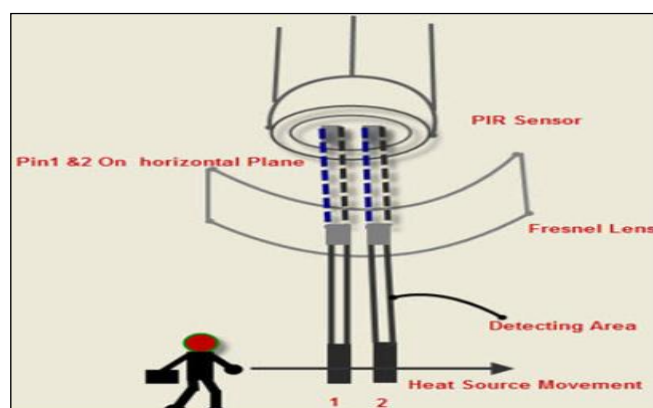


Figure 2-2 Passive infrared sensor

2.2.3 Piezoelectric sensor

These are the sensors which convert kinetic energy into electrical energy when subjected to any kind of vibrations or mechanical energy. These generate electrical signals whenever there is a change in its pressure. These sensors generate an electric charge that is proportionate to the amount of pressure applied. These sensors serve the purpose of not just telling about the passing of a vehicle, but much more additional information. We can get to know about various attributes like weight of the object, speed at which it is passing by, axle spacing and also the kind of vehicle it is by interpreting the above parameters. The disadvantages of the same would be that it requires multiple detectors to set up a proper instrumental location. Also, the sensors are extremely sensitive to high temperature and traffic stress.

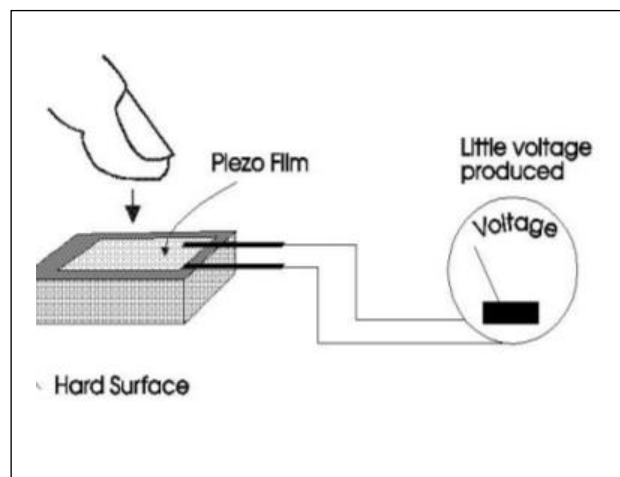


Figure 2-3 Piezoelectric sensor

2.2.4 Ultrasonic sensor

Ultrasonic sensors are the non-intrusive type of sensors. These are transducers (also known as transceivers), which work on the principle similar to sonar & radar systems and are used to determine the attributes of the vehicle based on the signals echoed back. These sensors can both transmit and receive the signals. These use high frequency sound waves as a medium, which are inaudible to human beings. The ultrasonic sensors send the signal, and wait for it to come back. The signal sent hits the object and bounces back, which is received by the sensors.

The received signals are interpreted, measuring the time interval between sending and receiving of the signal, and frequency of the received signal. These are then used to determine the presence of a vehicle, and also the distance of the vehicle. Despite its low power consumption and low cost realization, these are extremely sensitive to changes in environment, such as temperature, pressure, humidity etc.

Also these sensors must hit a hard surface to receive proper echoed signals.

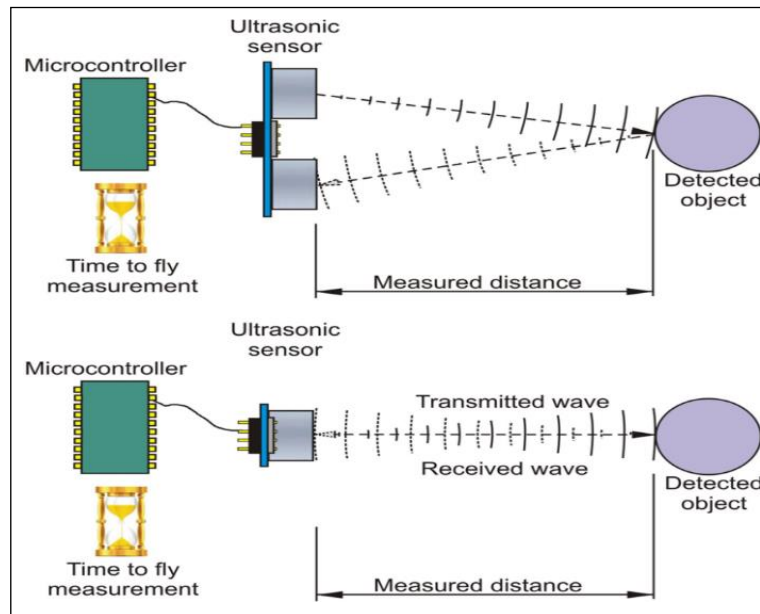


Figure 2-4 Ultrasonic sensor

Chapter # 3

Project Vision

3 Project Vision

3.1 Problem statement

These days, most of the office structures and shopping mall had built Underground parking and multilevel parking to overcome the number of cars which is increasing rapidly. Though, drivers are facing still difficulty to discover an available parking space to park their cars. The procedure of looking for a free parking space is time taking, puzzling and wasting fuel as well. Someone may miss or late for their important occasion. This situation may cause frustration for the drivers. Finally, the effect of deficient parking slot will cause the officer to have bad mood or customer to leave the shopping mall without buying anything. The side effect of this problem is thoughtful and need a better solution to handle it.

3.2 Problem background

Free parking slot these days is very limited for places like office building and shopping mall. There are many systems offer the vacant parking slot to the drivers. There are many flaws in existing systems to provide information about parking spaces to drivers. Thus, this system will provide the GUI which will show the available parking slot and the location of it. Moreover, the drivers will have a chance to pick the favorite parking. In this project, the wired sensor network will implement and operate with the ultrasonic sensors.

3.3 Proposed Solution

Proposed **“IoT based smart parking system”** is a based on NodeMCU and mobile application that is established to help drivers in searching available parking space during any time 24/7. The main goal of the System is to provide the Real Time occupancy status (empty/unavailable) of the parking spaces. The authorized user is permitted to select any date and can get the information related to parking spaces occupancy during that day. We divided our system into two parts, 1st part which is a software based mobile application and 2nd part is based on hardware.

3.3.1 Software

It is a mobile application which shows that has Real Time occupancy statuses of the parking slots. The application is refreshed when any changing occurs in parking space and the latest information is displayed for the user. This mobile application will be developed by using android software development kit tools (SDK). Application will use Maps application programming interface (API) provided by google to show the current status of parking spaces and to navigate the user to his destination or spot where user parked vehicle. The system will also use firebase real time database to send and receive data.

3.3.2 Hardware

The Physical part of System which is consist on NodeMCU board and ultrasonic sensors to detect the presence of objects in parking spaces it sends and receives Real Time data on firebase database. We use RFID tag and antenna in this system to recognize authorized user. System checks that user is Authorized or not, if user is unauthorized than parking gate will not open.

3.4 Project objectives

3.4.1 Business opportunity

Our system will provide business opportunity to parking owners. Everyone who has free space in front of their home or shop, they can give it to rent with the help of our system.

3.4.2 Improved parking

Clients locate the best Location accessible, saving time, assets and effort. The parking spots fill well and space can be utilized accurately.

3.4.3 Reduced traffic

Most of vehicles move around roads to find free parking space, our system will provide free parking space on the smart phones of drivers. So, they can find parking space before their arrival to particular place.

3.4.4 Reduced pollution

Searching for parking space consume thousands of barrels of fuel that causes pollution in atmosphere. With our system fuel consumption will be reduced hence pollution will also reduce.

3.4.5 Decreased Management Costs

Our system is IoT based in which sensors are involved to check free space rather than human involvement. Hence management cost will be decreased.

3.5 Project scope

These days, most of the office structures and shopping mall had built Underground parking and multilevel parking to overcome the number of cars which is increasing rapidly. Though, drivers are facing still difficulty to discover an available parking space to park their cars. The procedure of looking for a free parking space is time taking, puzzling and wasting fuel as well. Someone may miss or late for their important occasion. This situation may cause frustration for the drivers. Finally, the effect of deficient parking slot will cause the officer to have bad mood or customer to leave the shopping mall without buying anything. The side effect of this problem is thoughtful and need a better solution to handle it.

3.6 Constraints

- User must have android mobile phone for connecting to proposed system.
- Illiterate people can't use the system.
- Internet connection is compulsory to connect with the system.

Chapter # 4

Software Requirement Specification

4 Software Requirement Specification

4.1 List of Features

4.1.1 Registration

Initially, the user has to register his details with the application for the first time. This is a one-time registration. The user has to enter details like username, phone number and email-id. All this data will be stored on server.

4.1.2 Login

User can log into app by using his/her personal username and password. Most of the feature will be provided by system when user log into application. Without login, most of the feature will be not accessible for user.

4.1.3 Find Parking space

User can search for parking area on and around his/her destination. Select a particular parking area, browse through the various parking slots available in that parking area.

4.1.4 Select parking space

User is provided with numerous parking locations. User has to select one of the locations provided where he wants to park the vehicle.

4.1.5 Book a parking space

User reserve a parking lot by giving his/her RFID number. That specific slot is reserved for particular time.

4.1.6 Make payment

Parking charges will be calculated automatically based on expected time to stay at parking space. The charges for staying at parking area will be applied based on information provided by space owner. These charges will be paid by user to confirm his/her reservation on that particular parking area.

4.1.7 Users verification and authentication

On successful reservation, a confirmation page with user details and parking location is shown on User device. System will generate the Transaction ID for particular user. User will show RFID tag to system and he/she will be verified.

4.1.8 Navigation

Navigation is very important for that user who don't know about exact parking location. We will use API for manipulating maps provided by google. This API will access user's current location and his/her destination and then find the traveling route between current location and destination.

4.1.9 Notification

Here, User is received the notification from system. For example, in case of User over shoots its parking time, a notification would be sent to the user account. When an update is required for application the user will be notified.

4.1.10 Received bookings

This feature is for those users who have added their parking space or spaces to our application. Using this feature space owner can contact with client who wants to reserve slot in his/her parking space. Furthermore, space owner can approve parking lot for that client.

4.1.11 Check active session

User is supposed to reserve parking space for particular time period and after that allocated time an alert message will send to user and extra charges will be applied to user.

To overcome this problem, user have an option to check his active status and he/she can extend parking time.

4.1.12 Rent out parking space

Some people have extra space in front of their homes and some organizations have extra parking spaces and they want to use this extra space for public parking. They want to advertise

their parking for local business. They can add their extra parking lots in our system and our application user can also reserve parking in their parking area.

4.1.13 Account settings

Sometimes user wants to change his/her personal information for example add or remove debit/credit cards information. System will provide option for user to edit account details.

4.1.14 Show parking details

When a user is selected a parking area he/she wants to check parking details for example parking owner, parking security, parking type system will provide information to user.

4.2 Functional Requirements

“Smart parking system based on internet of things” is unique available solutions to at least lessen the traffic congestion produced due to finding parking spots. This system will provide information about the occupancy status of the spaces in the parking spot equipped by devices that notice the existence of vehicles. “Smart parking android application” will be connected to system with real time database and will show real time data received from system to user and will help drivers to make cool decisions which will lessen congestion.

4.2.1 Services

- Find Parking space
- Book a parking space
- Make payment.
- Navigation.
- Check parking status
- Rent out parking space

4.3 Non-Functional Requirements

4.3.1 Responsiveness

Spark application layout will respond accurately on almost all Android supported devices. For example, smartphones, tablets, smart LED's etc.

4.3.2 Efficiency

App will be efficient to use. This app will be used to reduce the real time problems so the speed of processing is very good to launch modules. It will be searched parking area nearest to your destination very efficiently. The basic logic is to use firebase instead of local server or cloud that gives more efficiency.

4.3.3 Maintainability

System will be easy to maintain because the way we have organized our code in manageable components and modules, therefore it will be to change the functionality in the future if needed. For this purpose, we will follow the basic methodologies of software engineering these are low coupling and high cohesion that allow us to manage our code.

4.3.4 Modularity

Modularity refers to the concept of making multiple modules first and then linking and combining them to form a complete system. In our system we will make separate classes or activities for our major use cases named as reservation, registration, or login etc., and then we will make some main classes these will reuse for other modules for example validation etc. and hence we will get reusability of code.

4.3.5 Usability

Usability of system will be very attractive because we use different types of animations and graphics in it. User-interface will attract the user to use it which increase the usability of the application.

4.3.6 Easy to Use

This android based application is for general public. So, we will make this app simple possible with minimum ambiguity.

4.3.7 Space Requirement

App will be able to use at any kind of android supported device such like inexpensive devices which have low memory or ROM, our application total memory consumption will be less than 30 MB's.

4.3.8 User Interface Specifications

The user interface will be kept simple so that users that are not as knowledgeable with computers can use this application.

4.3.9 Reliability

The reliability of the software design must be determined. The reliability of the software system refers to the ability to avoid fault occurred in the process of system running, as well as the ability to remedy troubles once the fault occurs.

4.3.10 Understandability

The understandability of software not only require clear and readable document, but the simplified structure of software itself, which requires the designer possess keen insight and creativity, and know well about the design objects.

4.3.11 Simple program

To keep the program simple and clear, good programmers can use simple program to solve complex problems.

Chapter # 5

High Level Use Cases

5 High Level Use Cases

5.1 Use case diagram.

Design is the initial phase in the development stage for any procedures and standards to define a device, a procedure or system in adequate detail to allow its physical acknowledgment. Once the software necessities have been breaking down and indicated the software design includes three specialized exercises designing, coding, implementation and testing that are required to construct and check the product.

The design exercises are of key significance in this stage, in light of the fact that in this movement, choices at last influencing the implementation of the product usage and its simplicity of maintenance are made. These choices have the last bearing upon dependability and viability of the system. Design is the best way to precisely make an interpretation of the client's necessities into completed software or a product.

5.1.1 System Use cases for Smart parking based on internet of things

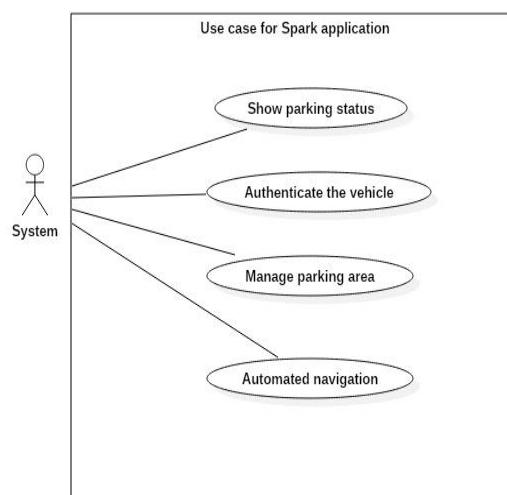


Figure 5-1 Smart parking system use case diagram

5.1.2 Use cases for unauthorized user

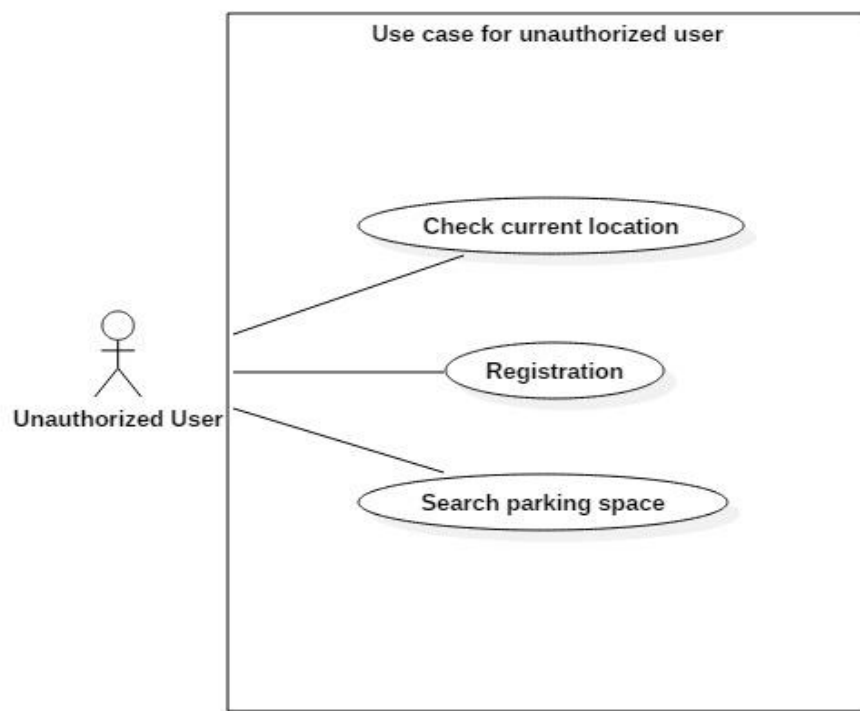


Figure 5-2 Use case diagram for unauthorized user

Chapter # 6

System Detailed Design

6 System detailed design

6.1 Use case of authorized user

A user is supposed to authorize if he/she has spark registration id that consist of RFID tag id. Using this ID user can access the entire features of spark application.

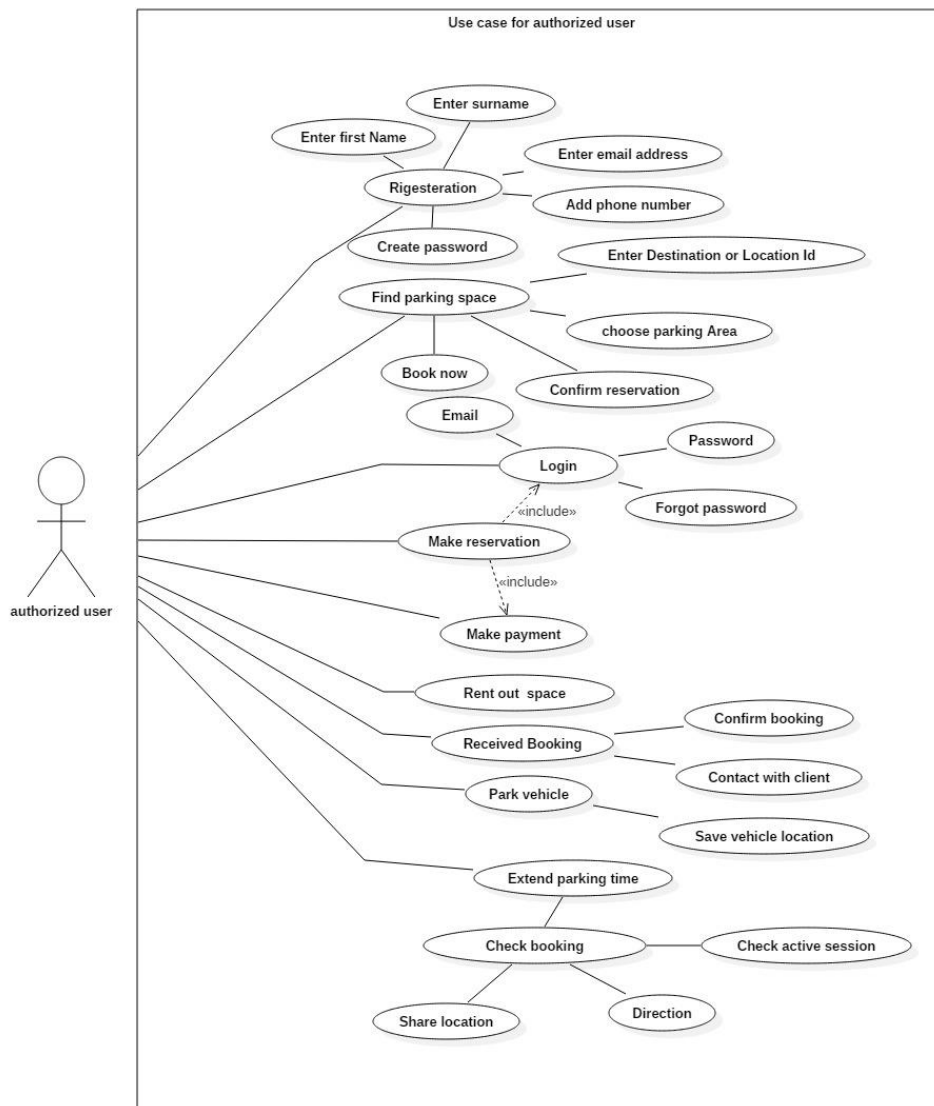


Figure 6-1 Detailed use case diagram for authorized user

6.2 Flowchart

A flowchart is a visual portrayal of the succession of steps and choices expected to play out a procedure. Each progression in the grouping is noted inside a diagram shape. Steps are connected by associating lines and directional arrows. This enables anybody to see the flowchart and consistently take after the procedure from start to finish. A flowchart is an intense business instrument. With appropriate plan and development, it conveys the means in a procedure successfully and productively.[8]

6.2.1 Overall Spark application flow chart

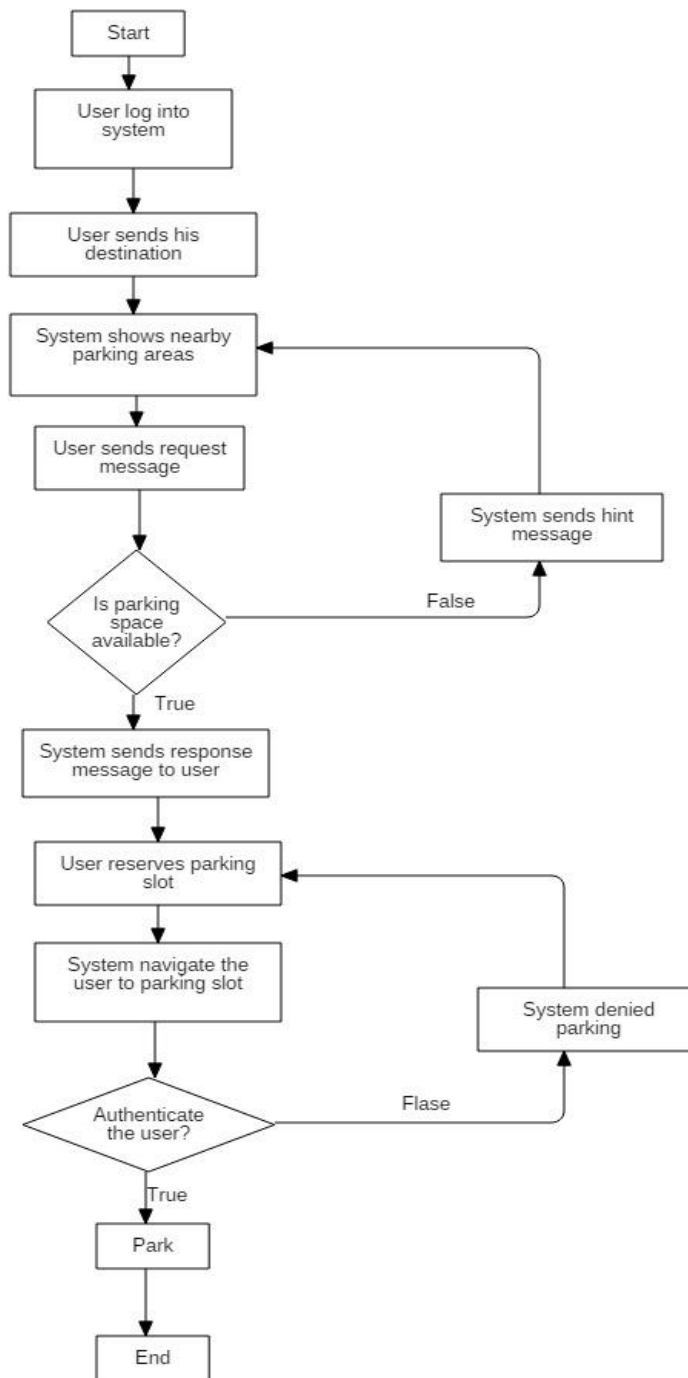


Figure 6-2 Spark application overall flow chart

6.2.2 Overall Spark hardware flow chart

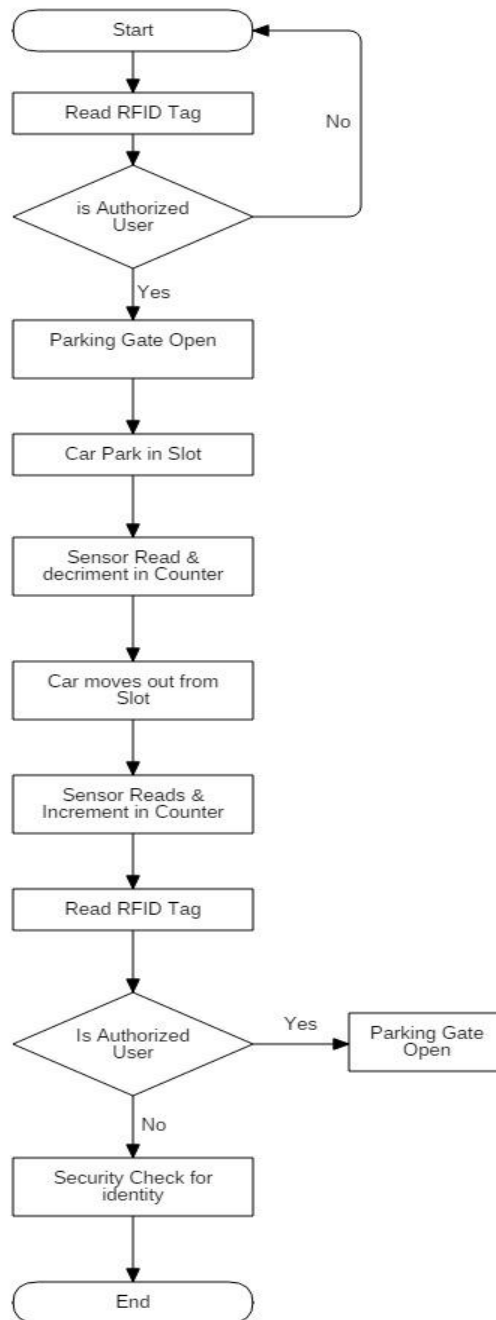


Figure 6-3 Spark hardware overall flow chart

6.3 Sequence diagram

A sequence diagram demonstrates object communications organized in time succession. It portrays the items and classes engaged with the situation and the arrangement of messages traded between the objects expected to complete the functionality of the situation. Sequence diagrams are usually connected with use case recognitions in the Logical View of the system being functioned on. Sequence diagrams are occasionally called event diagrams or event scenarios.

Sequence diagram Displays, as similar vertical lines (links), unlike procedures or objects that live instantaneously, and, for example horizontal arrows, the messages traded between them, in the request in which they happen. This permits the particular of straightforward runtime situations in a graphical way.[9]

6.3.1 Sequence Diagram of spark system

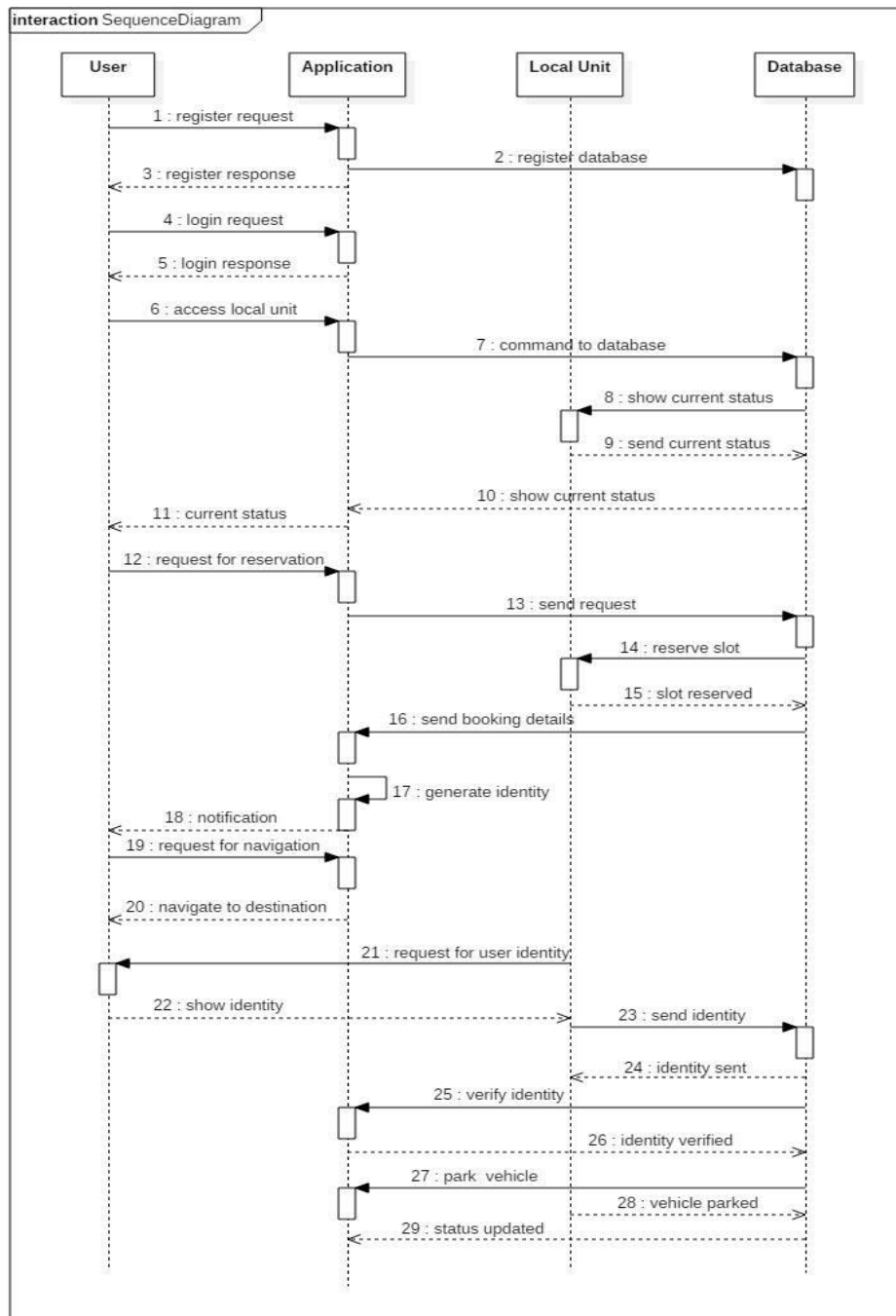


Figure 6-4 Sequence diagram for spark system

6.4 Hardware Prototype Diagrams

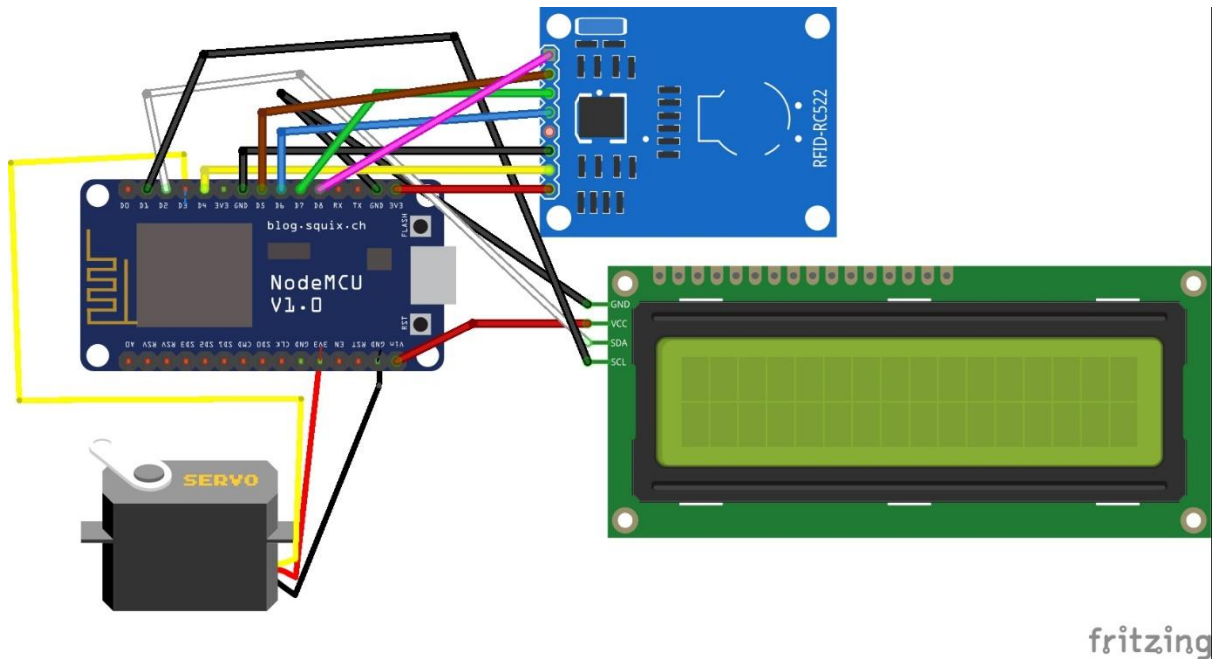


Figure 6-5 prototype diagram I

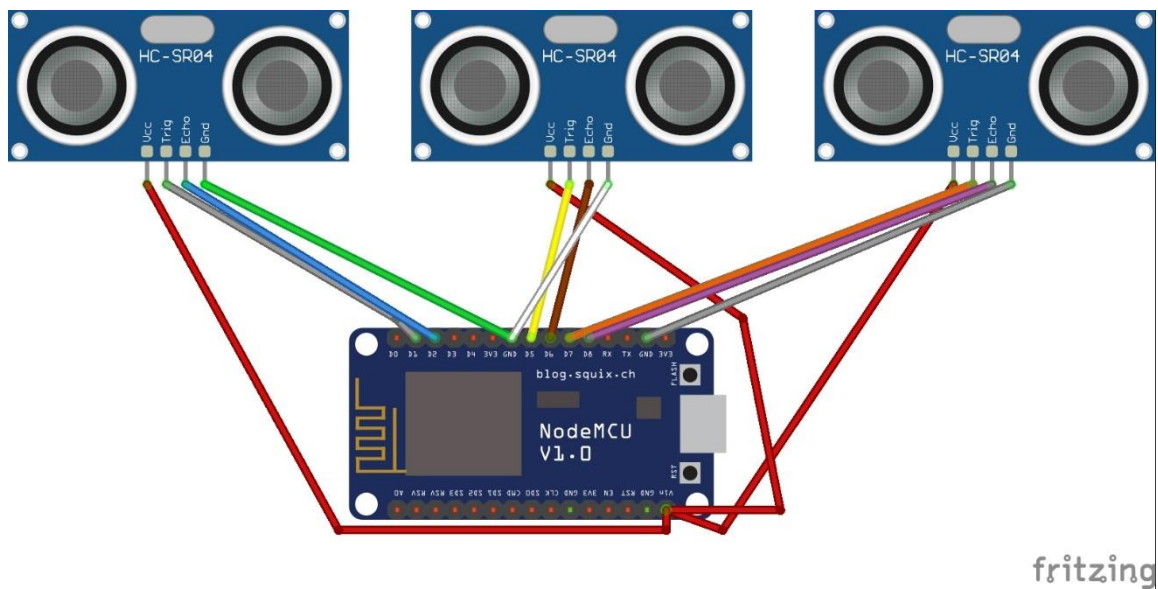


Figure 6-6 prototype diagram II

6.5 Hardware Circuit Diagrams

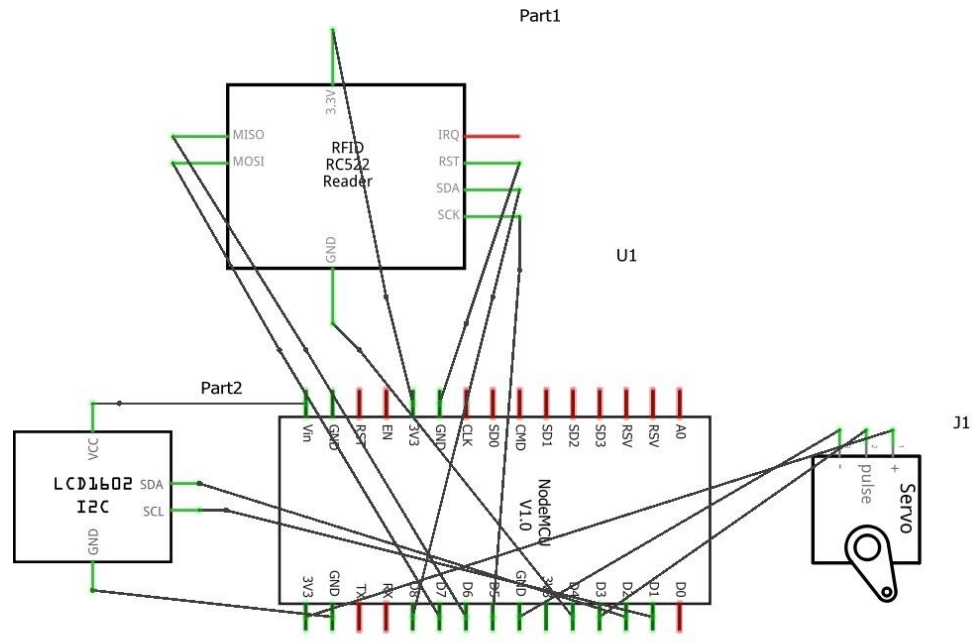


Figure 6-7 Circuit diagram I

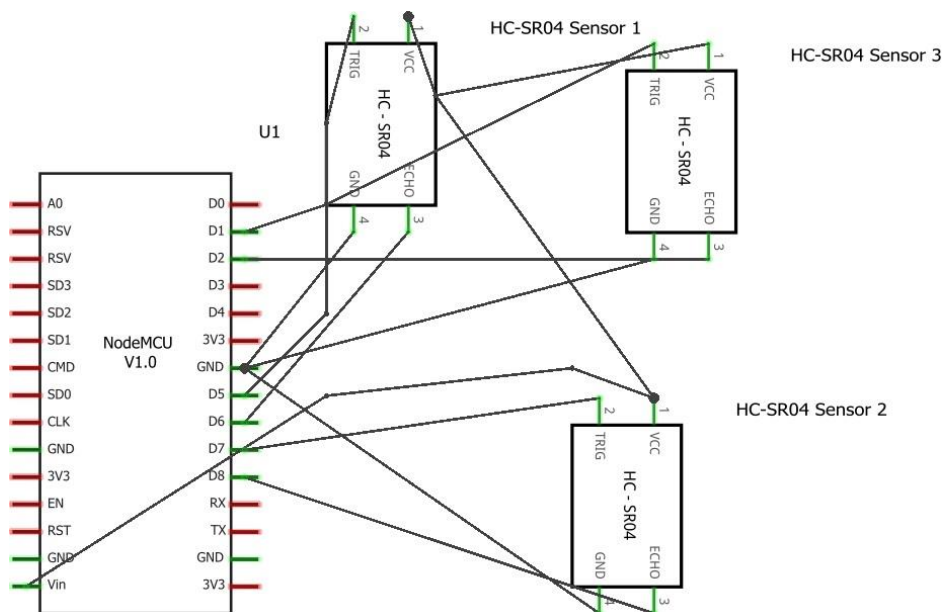


Figure 6-8 Circuit diagram II

Chapter # 7

Implementation

7 Implementation

7.1 Software

7.1.1 Software We Use

We use Android studio for the development of this application, because android is one of the most used platform especially in mobile or hand-held computers.

Android is an open source and Linux based working framework for cell phones, for example, cell phones and tablet PCs. Android was created by the Open Handset Alliance, driven by Google, and different organizations. Android programs depends on Java programming language so in the event that you have fundamental comprehension on Java programming then it will be an enjoyable to learn Android application development.[10]

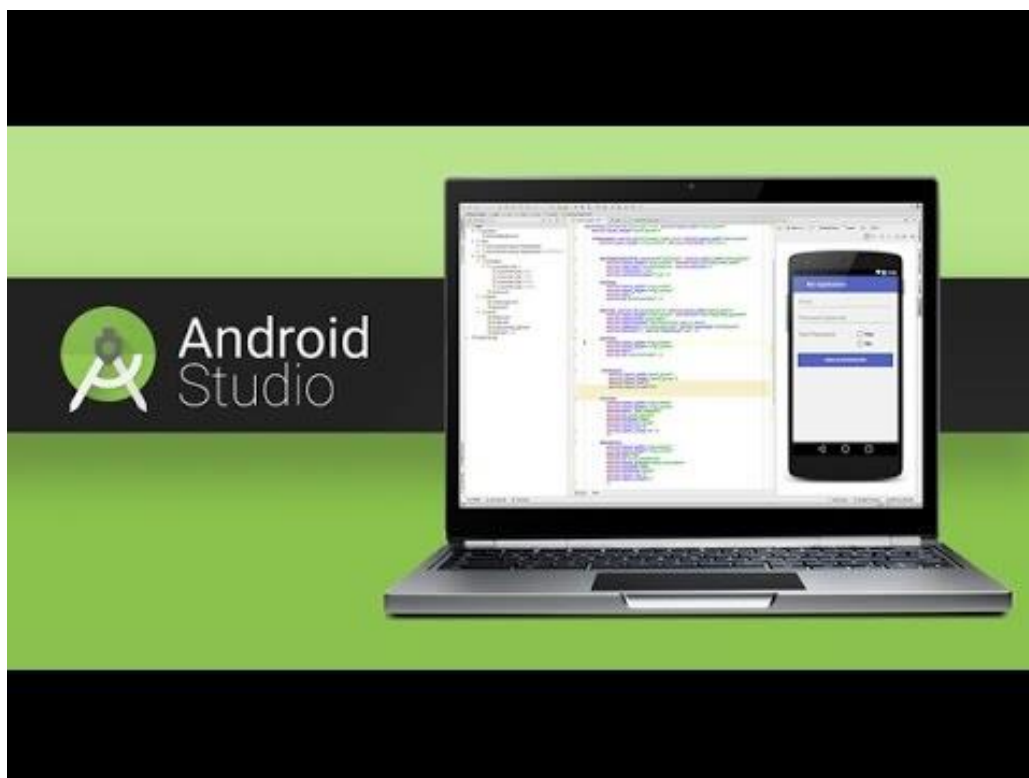


Figure 7-1 Android studio IDE

7.1.2 Build developing environment of Android

The applications of Android need to run based on Android environment. The following is the configuration requirement and installation steps of Android development environment.

- Operation system: Windows XP, Linux, Windows 7/8/10.
- Software Android SDK (Software Development Kit) ADT (Android Development Tool).
- JDK: Java Runtime Environment virtual machine Java Development Kit(JDK).

7.1.3 Installation steps of the developing environment

- Step 1: install the Java virtual machine JDK version – 6.
- Step 2: install android studio.
- Step 3: install the Android SDK: first download the Android SDK.

7.1.4 Why we use Android Platform?

We use Android platform for this assistive application because android is one of the emerging technology in computer science world. Android based devices are easy to use at anywhere you want. That's why if a user wants entertainment, then he will have an option to use the application anywhere.

Secondly, if any user wants to install or get access for application then android is also a better choice due to best accessibility Android App Store.

7.1.5 Firebase

Firebase in an expansive sense is a portable and web application improvement stage. Firebase liberates engineers to center creating fabulous client encounters. You don't have to oversee servers. You don't have to compose APIs. Firebase is your server, your API and your information store, all composed so blandly that you can change it to suit generally needs. No doubt, you'll every so often need to utilize different bits of the Google Cloud for your advanced applications. Firebase can't be everything to everyone. But it gets pretty close.[11]

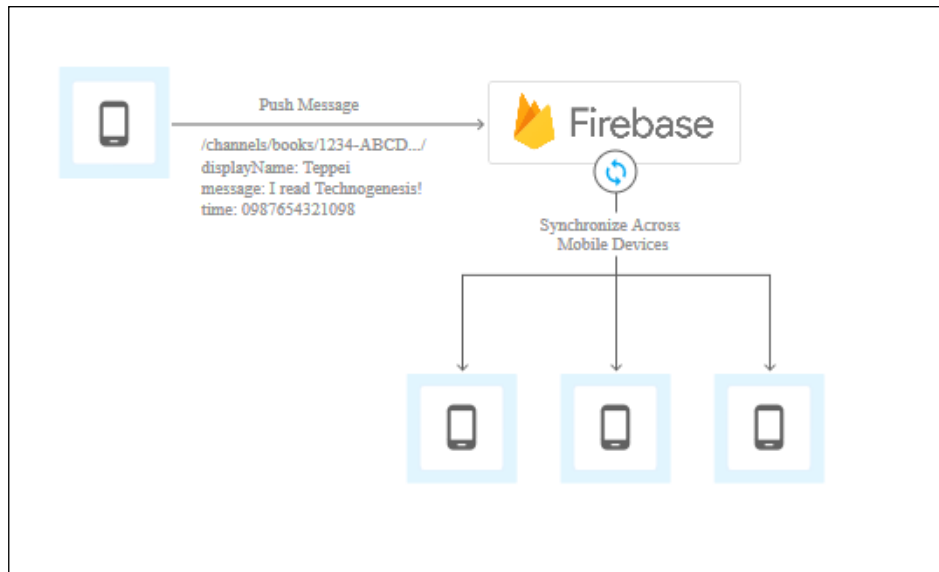


Figure 7-2 Firebase

7.1.5.1 Real-time database

Firestore is a real-time database that you can speak with specifically from the customer. The minute you spare your JSON information to Firestore, these progressions are instantly sent to all customers, of web and versatile, who asked for them. Firestore will enable you to assemble present day applications speedier than at any other time, with worked in static record facilitating, client administration, and security rules.

Nothing looks at to ongoing information. It is the method for what's to come. Most databases require you to make HTTP calls to get and match up your information. What's more, these databases give you information just when you request it. With Firestore, when you associate your application to it, you're not interfacing through typical HTTP. You're interfacing through a WebSocket. WebSockets are a considerable measure quicker than HTTP. You won't need to make individual WebSocket calls since one attachment association is all you require. The majority of your information synchronizes naturally through the single WebSocket as quick as your customer's system can convey it and Firestore sends you knew information as and when it's refreshed. At the point when your customer spares a change to the information, all customers who are associated, get the refreshed information in a split second. The Firestore constant database furnishes us with the capacity to spare, recover and match up our information with our NoSQL cloud database.

This information can be synchronized over all customers progressively. You see the Firebase constant database is a cloud-facilitated NoSQL database, where the information put away as a JSON is synchronized continuously to every associated customer. In this way, it has distinctive enhancements and functionalities contrasted with a social database.[12]

In our project we use firebase real time database to store all information about parking locations and we also connect this database to our local unit or hardware. We also store user's data in firebase database.

7.1.5.2 File Storage

Firebase Storage gives a straightforward method to save binary files most frequently pictures, yet it could be anything to Google Cloud Storage specifically from the customer. Firebase Storage has its own arrangement of security guidelines to shield your Cloud can from the majority, while giving nitty gritty compose benefits to your validated customers. We stored user images and parking images into file storage provided by firebase.

7.1.5.3 Authentication

Firebase authentication has a per-characterized in email/password confirmation framework. It additionally watches over Google, Facebook, Twitter and GitHub. We'll center around email/password authentication. Firebase's OAuth2 framework is all around perceived and generally copy/paste.

To authenticate our clients, we utilize firebase authentication module.

7.2 Hardware We Use

7.2.1 NodeMCU Microcontroller

NodeMCU is an open source IoT stage. it incorporates firmware which keeps running on the esp8266 wi-fi soc from espressif frameworks, and equipment which depends on the esp-12 module. the expression "nodemcu" as a matter of course alludes to the firmware as opposed to the dev packs. The term "NodeMCU" by evasion exemplifies to the firmware in place of the development kits. The firmware usages the Lua scripting language. It depends on the eLua venture, and arranged on the Espressif Non-OS SDK for ESP8266.

It uses many open source ventures, for example, lua-cjson, and spiffs.

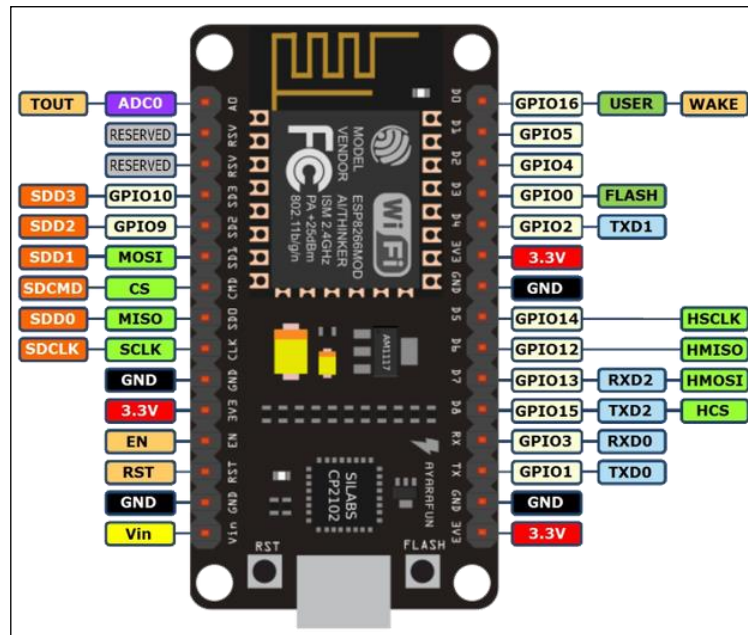


Figure 7-3 NodeMCU microcontroller

7.2.2 HC-SR04 ultrasonic sensor

The HC-SR04 ultrasonic sensor uses sonar to find distance to an object like bats do. It provides outstanding non-contact range recognition with high correctness and stable readings in an easy-to-use package from 2cm to 400 cm or 1" to 13 feet. Its procedure isn't influenced by daylight or dark material like Sharp rangefinders are (albeit acoustically delicate materials like fabric can be hard to recognize). It comes complete with ultrasonic transmitter and receiver module.



Figure 7-4 HC-SR04 Ultrasonic Sensor

7.2.3 Servomotor

A servomotor is a rotating actuator or straight actuator that grants for correct control of precise or direct position, velocity and acceleration. It contains a suitable motor attached to a sensor for position feedback. It also needs a relatively refined controller, often a devoted module designed specifically for use with servomotors. Servomotors are not exact class of motor although the term servomotor is often used to refer to a motor appropriate for use in a closed-loop control system. Servomotors are utilized in applications, for example, mechanical autonomy, CNC apparatus or robotized producing.



Figure 7-5 Servomotor

7.2.4 LCD for display

Liquid crystal displays (LCDs) are a generally used to display data in devices such as calculators, microwave ovens, and numerous other electronic devices.

The LCDs have a parallel interface, meaning that the microcontroller has to operate several interface pins at once to control the display.



Figure 7-6 LCD for display

7.2.5 RFID System

Radio-Frequency Identification (RFID) is the use of radio waves to read and capture information stored on a tag attached to an item. A tag can be read from up to numerous feet away and does not need to be within direct line-of-sight of the reader to be tracked.

A RFID system is made up of two parts: a tag or label and a reader. RFID tags or labels are fixed with a transmitter and a receiver. The RFID module on the tags have two parts: a microchip that stores and processes data, and an antenna to receive and transmit a signal. The tag contains the specific serial number for one specific item.

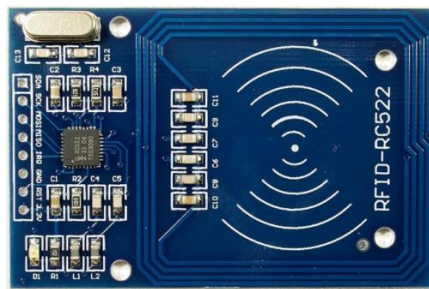


Figure 7-7 RFID system

7.2.6 Wi-Fi Module

The ESP8266 Wi-Fi module is an entire Wi-Fi network where you can without much of a stretch associate as a serving Wi-Fi connector, remote web access to interface to any microcontroller manufactured design on its basic connectivity through Serial Communication or UART interface.

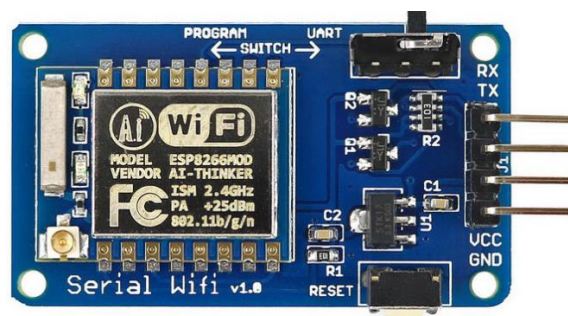


Figure 7-8 Wi-Fi module

7.2.7 I2C LCD

I2C_LCD is an easy-to-use display module it can make display easier. Utilizing it can lessen the trouble of make, with the goal that producers can center around the center of the work. We built up the Arduino library for I2C_LCD, client simply require a couple of lines of the code can accomplish complex graphics and text show features. It can supplant the serial screen of Arduino in some place, you can get running data without a PC. More than that, we likewise build up the devoted picture information change over programming (bitmap converter) now is accessible to help PC stage of windows, Linux, Mac OS. Through the bitmap convert software, you can get your favorite picture displayed on I2C_LCD, without the need.

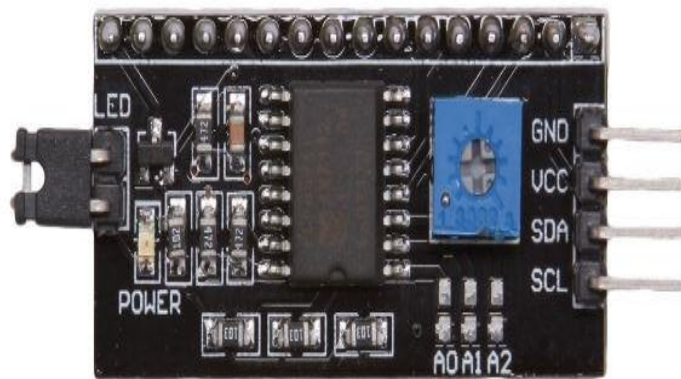


Figure 7-9 I2C LCD

7.3 Front end designing using XML

The frontend of an application is unmistakably human. It's what the client sees, contacts and encounters. The frontend of an application is less about code and more about how a client will decipher the interface into an ordeal. The general Programming languages associated with outlining the front end is XML.

Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Initially designed to address the difficulties of large scale electronic distributing, XML is likewise assuming an inexorably essential part in the trading of a wide assortment of information on the Web and elsewhere.

7.3.1 Components of front end designing

Following components are used in overall design of our (Spark) application.

7.3.1.1 Layouts

A layout specifies the structure for a UI in application, for example, in an activity. All components in the design are constructed utilizing a progressive system of View and ViewGroup objects. A View for the most part draws something the client can see and cooperate with. Though a ViewGroup is an unseen container that specifies the layout structure for View and other ViewGroup objects.

7.3.1.2 Notifications

A notification is a message that Android shows outside application's UI to furnish the client with updates, correspondence from other individuals, or other convenient data from your application. Clients can tap the notification to open application or make a move specifically from the notice.

7.3.1.3 Toasts

A toast gives straightforward input around an activity in a little popup. It just fills the measure of space required for the message and the present activity stays visible and interactive. Toasts naturally vanish after a timeout.

7.3.1.4 The app bar

The application bar, also called the activity bar, is a standout among the most vital outline components in application's exercises, since it gives a visual structure and intuitive components that are well-known to clients. Utilizing the application bar makes your application reliable with other Android applications, enabling clients to rapidly see how to work your application and have an incredible ordeal.

7.3.1.5 Dialogs

A dialog is a little window that prompts the client to settle on a choice or enter extra data. A dialog does not fill the screen and is routinely used for modular events that anticipate that

customers will make a move before they can proceed. Examples of dialogs DatePickerDialog or TimePickerDialog.

7.3.1.6 Menus

Menus are a typical UI part in numerous kinds of applications. To give a recognizable and steady user experience, you should utilize the Menu APIs to display user actions and different choices in your activities.

7.4 Back end development using JAVA

Android applications can be developed using Kotlin, Java, and C++. The Android SDK tools compile your code with any information and asset records into an APK, an Android bundle, which is a file document with an apk postfix. One APK record contains every one of the substance of an Android application and is the document that Android-powered devices use to install the application.

Every Android application lives in its own particular security sandbox, ensured by the accompanying Android security highlights:

- The Android working framework is a multi-client Linux framework in which each application is an alternate client.
- By default, the framework assigns each application Linux client ID (the ID is utilized just by the framework and is known to the application). The framework sets permissions for every one of the files in an application with the goal that exclusive the client ID allocated to that application can get to them.
- Each process has its own particular virtual machine (VM), so an application's code keeps running in disengagement from different applications.
- By default, each application keeps running in its own particular Linux process. The Android framework begins the procedure when any of the application's parts should be executed, and after that close down the procedure when it's never again required or when the framework must recover memory for different applications.

The Android framework implements the principle of least privilege. That is, each application, as a matter of course, approaches just to the parts that it requires to do its work and no more.

This makes an extremely secure condition in which an application can't get to parts of the framework for which it isn't given consent. Nonetheless, there are ways for an application to impart information to different applications and for an application to access system services:

- It's feasible to group two applications to have a same Linux client ID, in which case they can get to each other's files. To take care of system resources, applications with a similar client ID can also arrange to keep running in a similar Linux process and offer the same VM. The applications should likewise be marked with a similar certificate.
- An application can ask for permission to get to device information, for example, the client's contacts, SMS messages, the mountable stockpiling (SD card), camera, and Bluetooth. The client needs to expressly concede these permissions.

7.4.1 Components of backend development

Application parts are the basic building blocks of an Android application. Every segment is a passage point through which the framework or a client can enter your application. A few segments rely upon others.

There are four different types of app components:

7.4.1.1 Activities

An activity is the starting point for connecting with the client. It shows a solitary screen with a UI. For instance, an email application may have one activity that demonstrates a list of new messages, another activity to make an email, and another activity for perusing messages. Despite the fact that the activities cooperate to form a strong user experience in the email application, everyone is autonomous of the others. In that capacity, an alternate application can begin any of these activities if the email application permits it. For instance, a camera application can begin the activity in the email application that creates new mail to enable the client to share a photo. An action encourages the accompanying key interactions between system and application:

- Keeping track of what the client as of now thinks about (what is on screen) to guarantee that the system continues running the procedure that is facilitating the activity.

- Knowing that already utilized procedures contain things the client may come back to (halted activities), and in this way more exceedingly organize keeping those procedures around.
- Helping the application handle having its procedure slaughtered so the client can come back to exercises with their past state reestablished.
- Providing a route for applications to actualize client streams between each other, and for the system to arrange these streams.

7.4.1.2 Services

A service is a broadly useful section point for keeping an application running out of sight for a wide range of reasons. It is a part that keeps running out of sight to perform long-running activities or to perform work for remote procedures. A service does not give a UI. For instance, a service may play music out of sight while the client is in an alternate application, or it may get information over the system without blocking client cooperation with an activity.

Another part, for example, an activity, can begin the service and let it run or tie to it so as to associate with it. There are really two exceptionally particular semantics services inform the system concerning how to deal with an application: Started services advise the system to keep them running until the point that their work is finished. This could be to adjust a few information out of sight or play music even after the client leaves the application. Matching up data out of sight or playing music additionally speak to two unique kinds of began services that alter how the system handles them:

- Music playback is something the client is specifically mindful of, so the application tells the system this by saying it needs to be closer view with a notice to educate the client concerning it; for this situation the system realizes that it should make a decent attempt to keep that administration's procedure running, on the grounds that the client will be miserable in the event that it leaves.
- A regular background service is not something the user is directly aware as running, so the system has more freedom in managing its process. It may allow it to be killed (and then restarting the service sometime later) if it needs RAM for things that are of more immediate concern to the user.

Bound services run since some other application (or the system) has said that it needs to make utilization of the service. This is fundamentally the service giving an API to another procedure. The system in this manner knows there is a reliance between these procedures, so if process A is bound to a service in process B, it realizes that it needs to keep process B (and its service) running for A. Further, if process A is something the client thinks about, at that point it additionally knows to regard process B as something the client likewise thinks about. As a result of their adaptability (regardless), services have ended up being an extremely helpful building obstruct for a wide range of more elevated amount system ideas. Live backdrops, warning audience members, screen savers, input strategies, open services, and numerous other center system highlights are altogether worked as services that applications actualize and the system ties to when they ought to run.

7.4.1.3 The manifest file

Before the Android system can begin an application component, the system must realize that the component exists by perusing the application's manifest file, `AndroidManifest.xml`. Your application must announce every one of its parts in this file, which must be at the foundation of the application venture registry.

The manifest file shows a number of things to declaring the app's components, for example:

- Recognizes user's permissions the app is required, such as Internet or write-access to the gallery.
- Announces the minimum API Level compulsory by the application, centered on which APIs the application uses.
- States hardware and software features used or required by the application, such as NFC, Wifi services, or a camera.
- Declares API libraries the application requirements to be related against (other than the Android framework APIs), such as the Google Firebase library.

7.4.1.4 Permissions

The persistence of a permission is to shield the secrecy of an Android client. Android applications must appeal permission to access sensitive user data (such as Gallery and Mails), as well as certain system features (such as Bluetooth and multi touch screen).

Reliant on the feature, the system might allowance the permission mechanically or might rapid the user to favor the request.

Chapter # 8

Testing and Results

8 Testing and Results

Software testing is necessary for a project. Testing is made so as to evaluate and get the result for developed software. Software testing is made in a controlled form with all normality's and abnormalities deals with the system.

In this project the issues we faced in the beginning has overcome by us and have done test on them, got the expected results now. Other than these all the functions were working properly.

Following are the test cases which we have done for this project and the output we got.

Test No	Test case	Expected Result	Actual Result	Pass/Fail
1.	Check phone number and email formats	The system should not accept invalid email or phone number.	Same as Expected	Pass
2.	Check Password length	Password length should not less than eight characters.	It wasn't accepting the password less than eight.	Pass
3.	Check permissions	Application should not be running if permissions are not allowed by user.	Same as Expected	Pass
4.	Find current location	The system should find current location.	System was not able to find current location sometimes	Fail
5.	Check current balance	Reservation should not done if balance is less than calculated charges for parking	Same as Expected	Pass
6.	Change of password	It requires old password	It was working properly.	Pass
7.	Verify email address	System should send email to provided email address	It was working properly	Pass

8.	Send Notifications	If user has new booking system should send notification	Same as Expected	Pass
9.	Navigate to destination	Application should navigate user to his/her destination	Same as Expected	Pass

Table 3 Testing and results

Chapter # 9

Conclusions and Future Work

9 Conclusions and future work

We started working on this project keeping some specific requirements in our mind. We have confidently achieved our set goals and finally we develop a IoT based smart parking system. Using this system user can find free parking space in a city and book parking where this system is deployed. User has also provided feature to pay parking fee with application.

In future we will work on quality of system and some other features that will be provided to user in future.

- First of all, we will work on save current location where car has parked in parking area and user will have option to share this location with his/her friends.
- Our next step will be on indoor maps that will help full for users to navigate them to exact location of parking spot where they parked car.
- We are looking trust full payment method available in Pakistan as soon as possible we will integrate it to our system for online payments now we are using only our own payment method that is not online.

Chapter # 10

User Manual

10 User Manual

10.1 Software

Here's a user guide assisted with screenshots of the actual scenarios that a user might face.

10.1.1 Starting application

This is starting screen of Spark application that has two options.

- Create account with Email
- Login

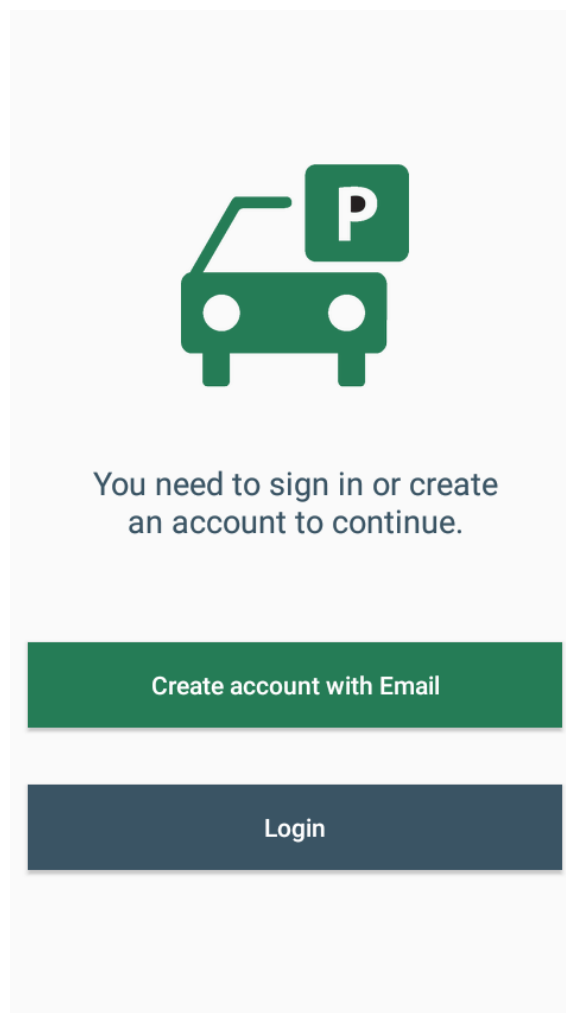
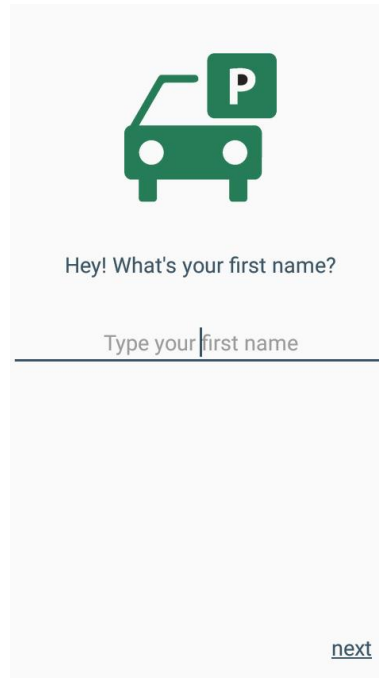


Figure 10-1 Starting screen of Spark application

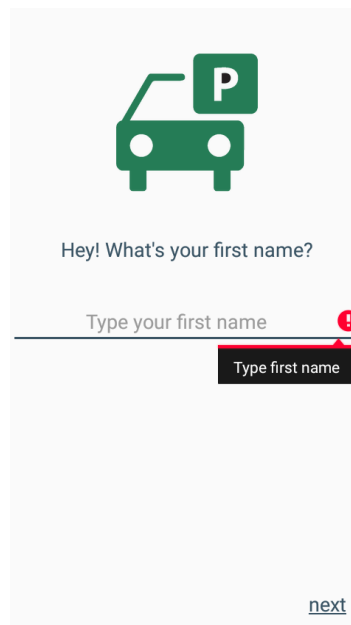
10.1.2 Create account with Email

After clicking on button create account with Email this screen will open. You have to fill following information shown in figures.



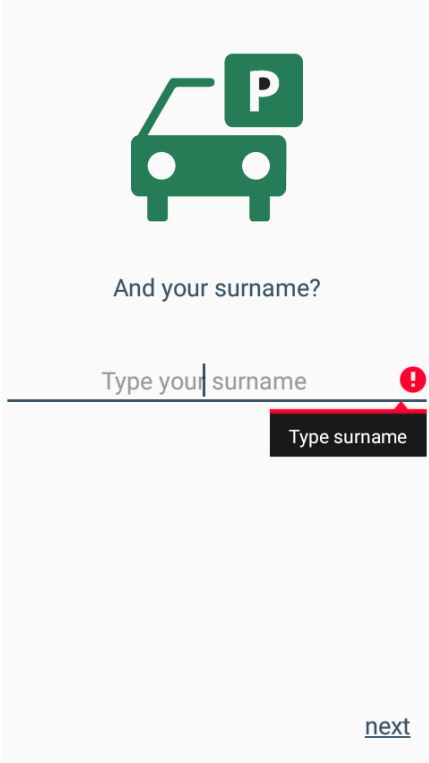
The image shows a mobile application screen for creating an account. At the top, there is a green icon of a car with a 'P' parking sign. Below the icon, the text 'Hey! What's your first name?' is displayed. Underneath, there is a text input field with the placeholder text 'Type your first name'. At the bottom right of the screen, there is a blue link labeled 'next'.

Figure 10-2 First name input screen



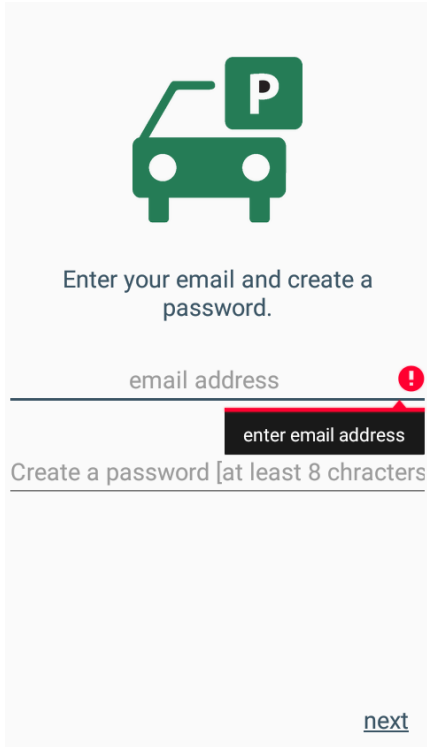
The image shows the same mobile application screen as Figure 10-2, but with an error message. The text input field is now empty, and a red exclamation mark icon is visible to the right of the field. Below the input field, there is a black rectangular box with the text 'Type first name' in white. At the bottom right of the screen, there is a blue link labeled 'next'.

Figure 10-3 Error message for empty first name




The screen features a green car icon with a 'P' parking sign above it. Below the icon, the text 'And your surname?' is displayed. A text input field contains the placeholder 'Type your|surname'. To the right of the input field is a red exclamation mark icon. A black tooltip with the text 'Type surname' is positioned below the input field. At the bottom right, there is a blue 'next' link.

Figure 10-4 Second name input screen



The screen features a green car icon with a 'P' parking sign above it. Below the icon, the text 'Enter your email and create a password.' is displayed. There are two input fields. The first field is labeled 'email address' and has a red exclamation mark icon to its right. A black tooltip with the text 'enter email address' is positioned below this field. The second field is labeled 'Create a password [at least 8 chracters]'. At the bottom right, there is a blue 'next' link.

Figure 10-5 Input screen for email address and password




Enter your email and create a password.

test@gmail.com

.....

[next](#)

Figure 10-6 Filled email and password screen



Add your phone number

+92341003949

[Why do we need this?](#) [finish](#)

Figure 10-7 Input screen for valid phone number

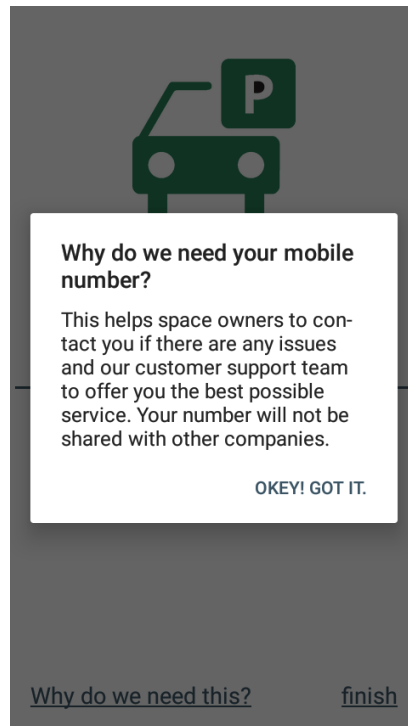


Figure 10-8 Why we need your phone number screen

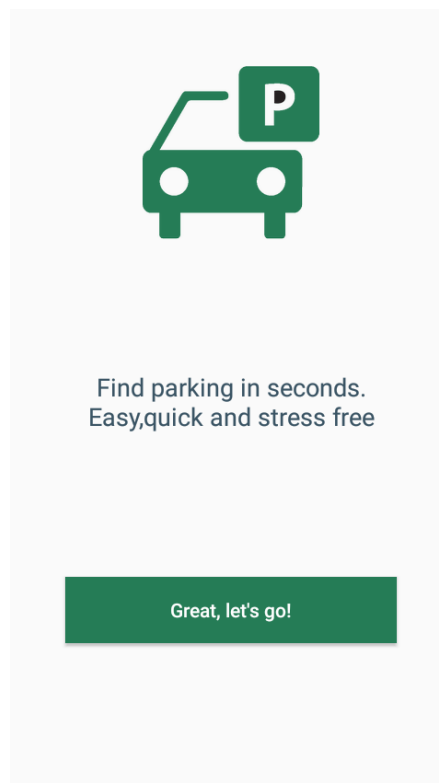


Figure 10-9 Welcome screen of Spark application

10.1.3 Permissions

There are some permissions you have to allow to Spark application to use its cool features.

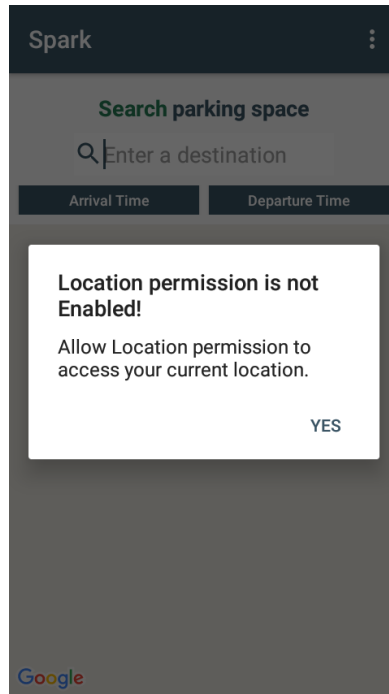


Figure 10-10 Location permission screen

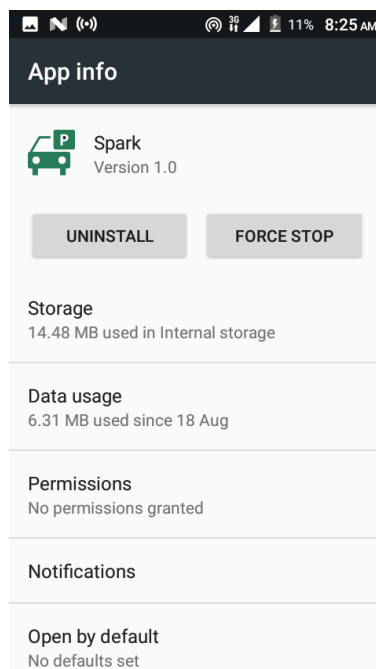


Figure 10-11 Application information screen

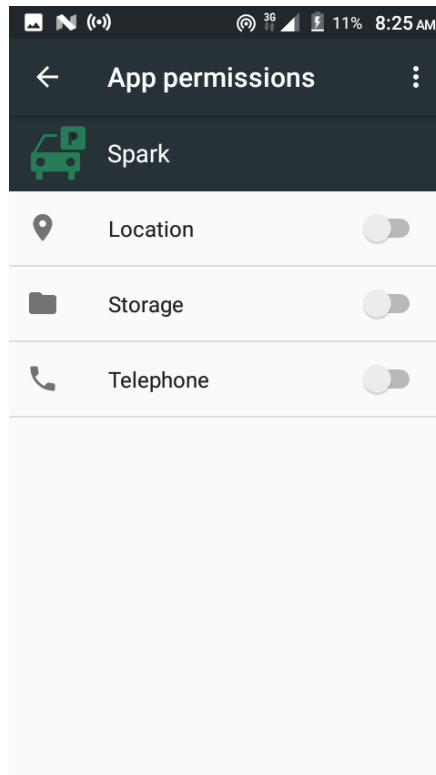


Figure 10-12 Permissions screen

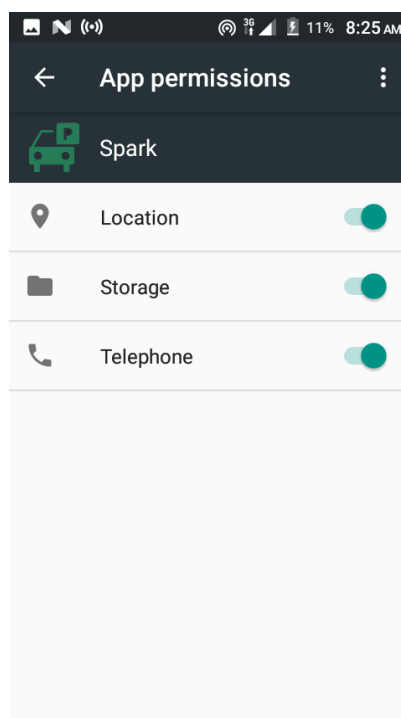


Figure 10-13 Allowed permissions screen

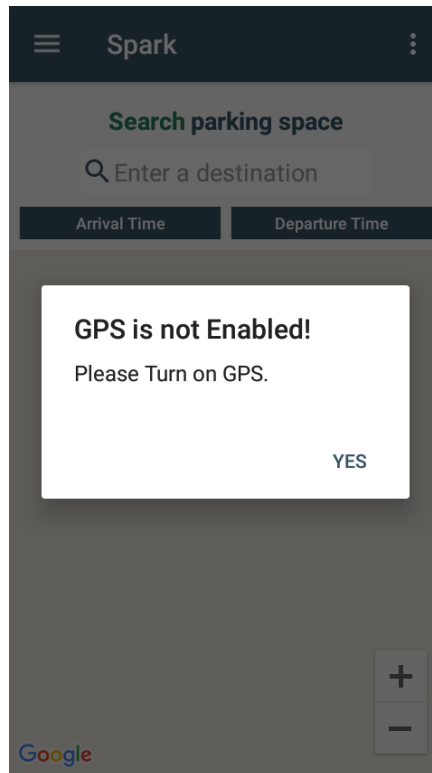


Figure 10-14 Enable GPS screen

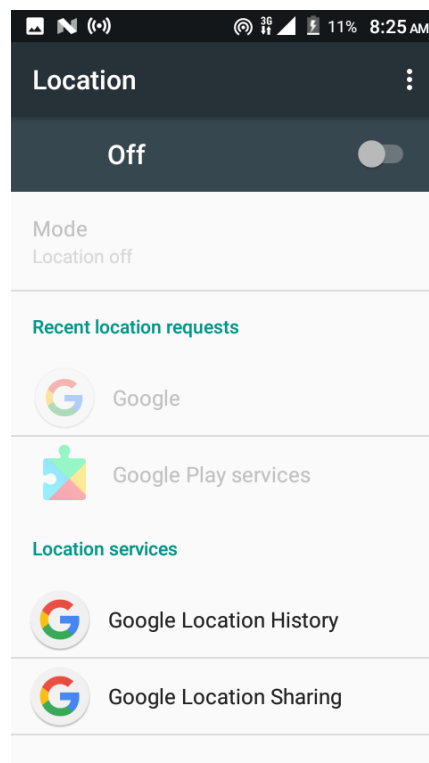


Figure 10-15 Location settings screen

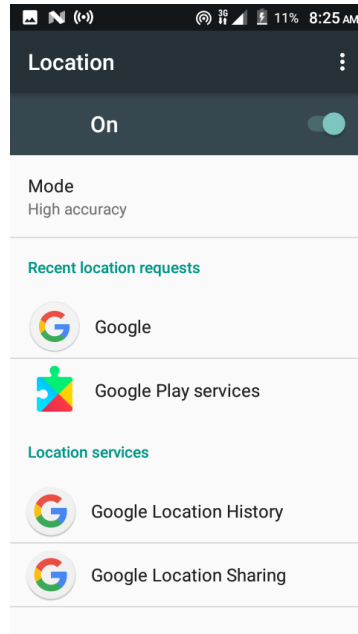


Figure 10-16 Enabled GPS screen

10.1.4 Application navigation drawer

To navigate between different screens, you are provided with a cool navigation drawer.

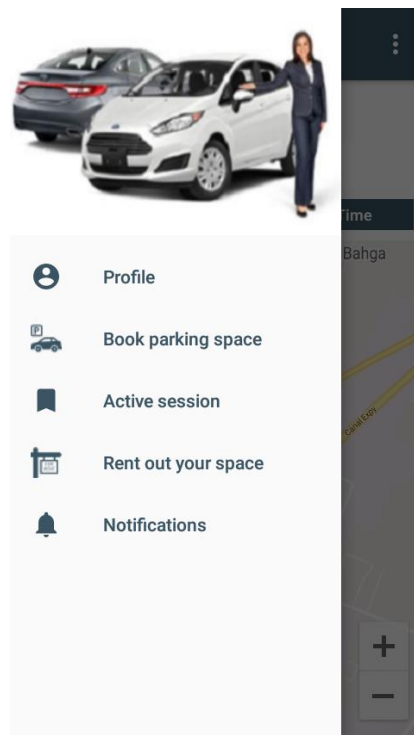


Figure 10-17 Navigation drawer screen

10.1.5 Profile settings

After successful registration you can change your information.

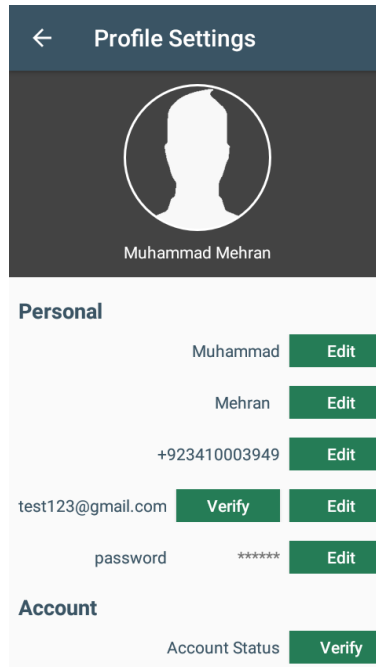


Figure 10-18 Profile settings screen

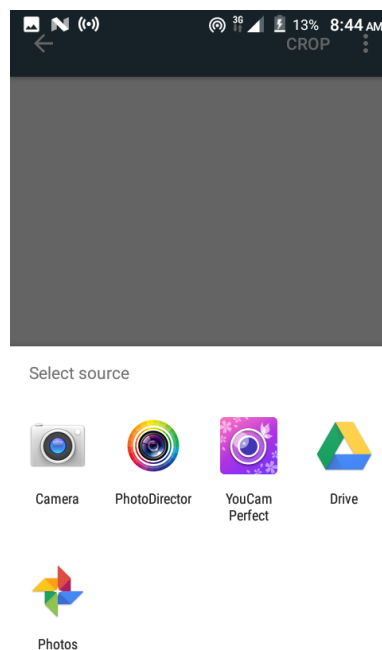


Figure 10-19 Select image from phone screen

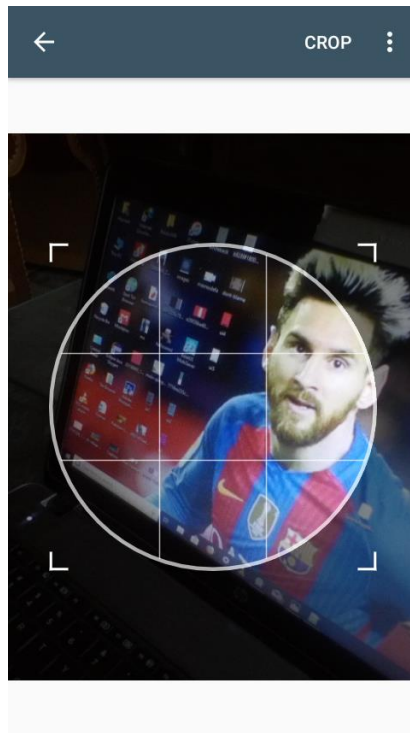


Figure 10-20 Crop, rotate, flip profile photo screen

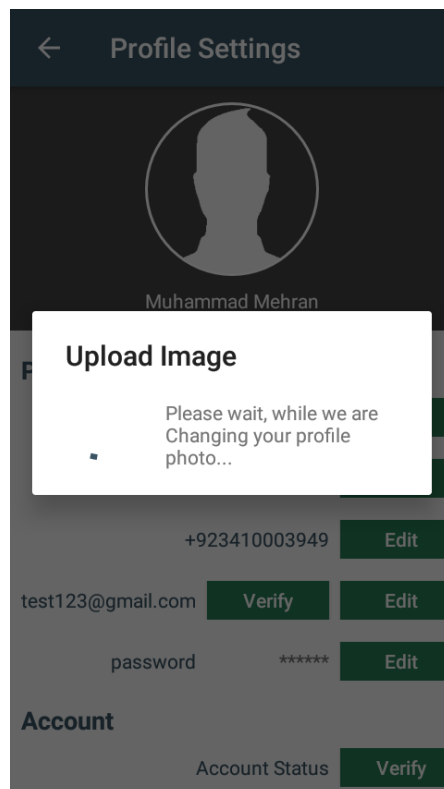


Figure 10-21 Uploading profile photo screen

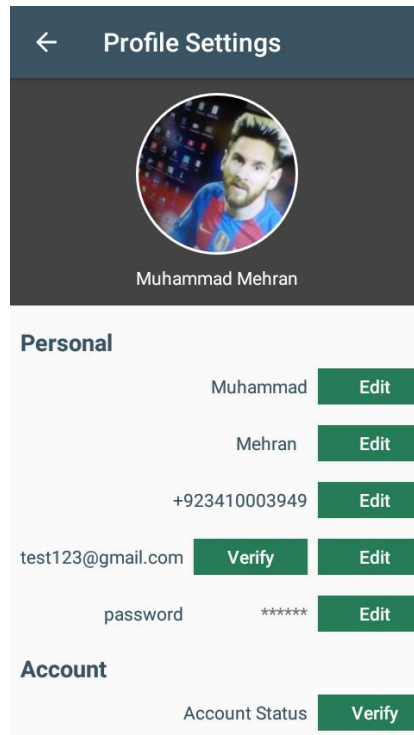


Figure 10-22 Changed profile photo screen

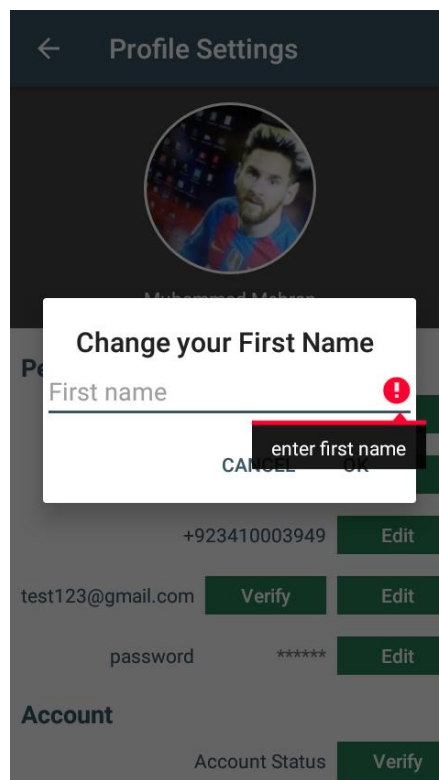


Figure 10-23 Edit first name screen

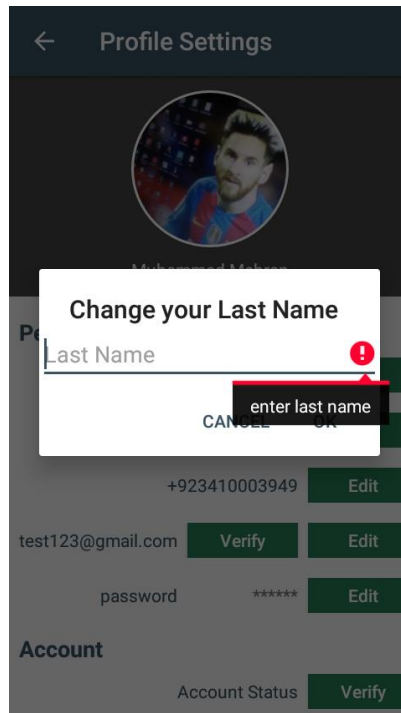


Figure 10-24 Edit last name screen

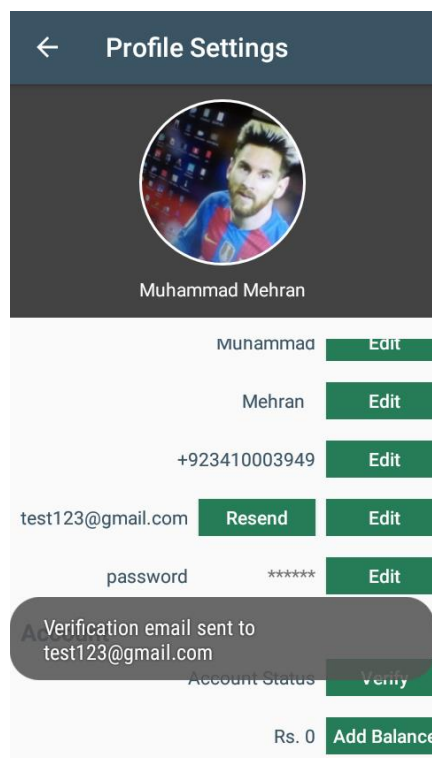


Figure 10-25 Verify email address screen

10.1.6 Account verifications

To use reservation feature you have to verify your account with following steps.

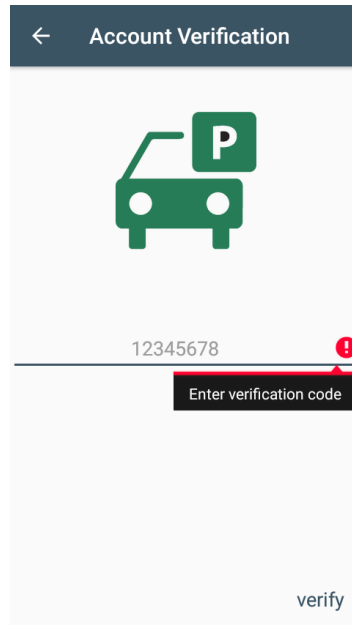


Figure 10-26 Account verification screen

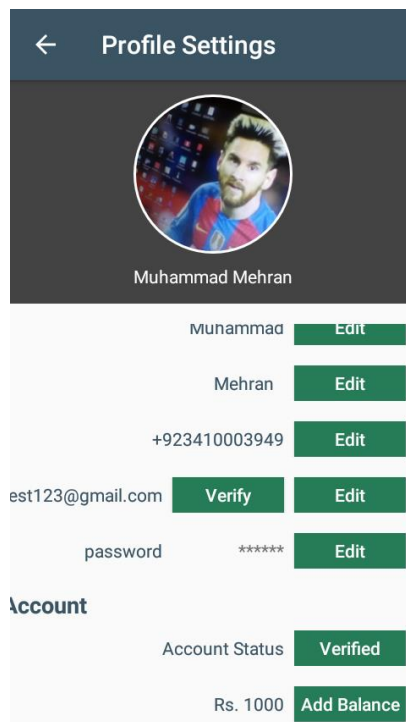


Figure 10-27 Verified account screen

10.1.7 Search parking space

Main screen of application provides you a search field by using it you can easily find parking spaces around your destination.

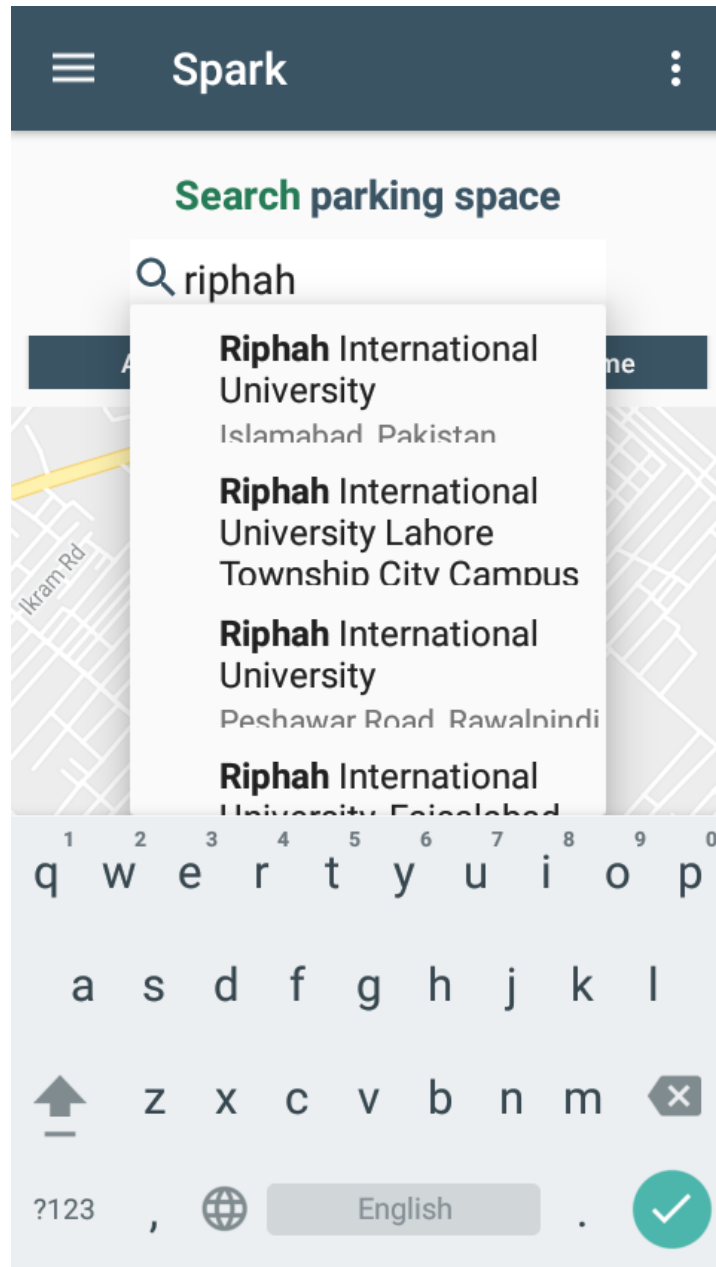


Figure 10-28 Search parking space screen

10.1.8 Reservation

After searching free space next step is to reserve free slot in your desired parking area following screen shots will help you to reserve slot.

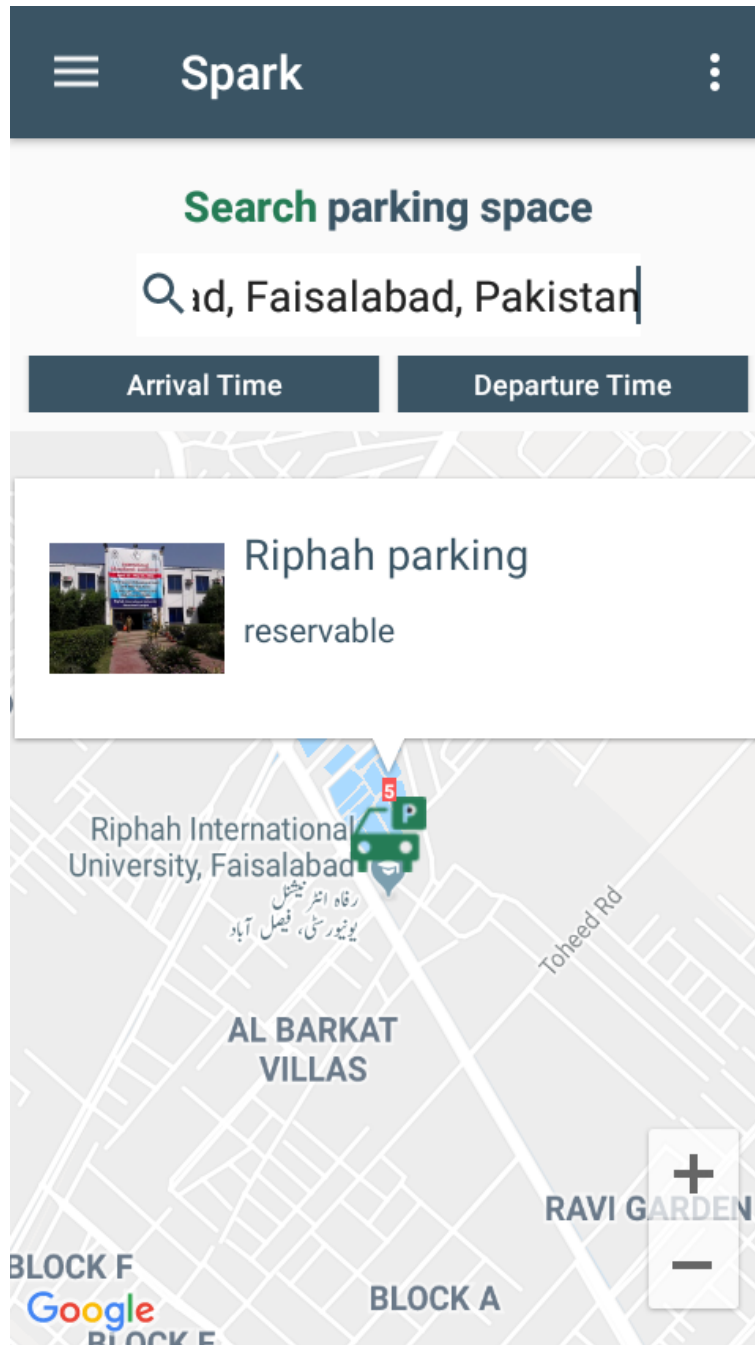


Figure 10-29 Selected parking area screen

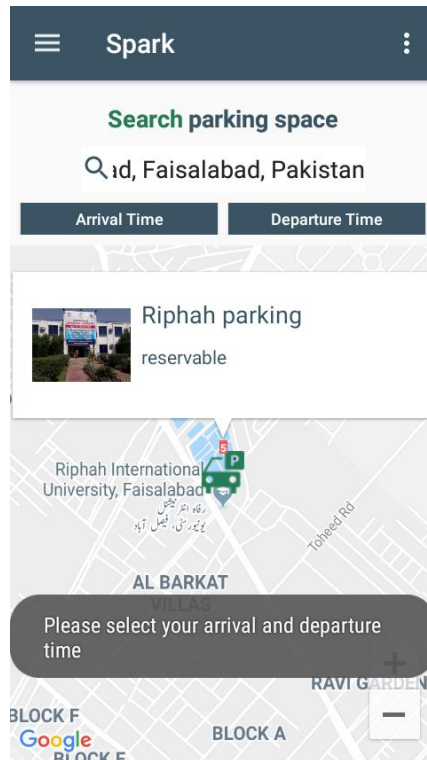


Figure 10-30 Select arrival and departure time screen

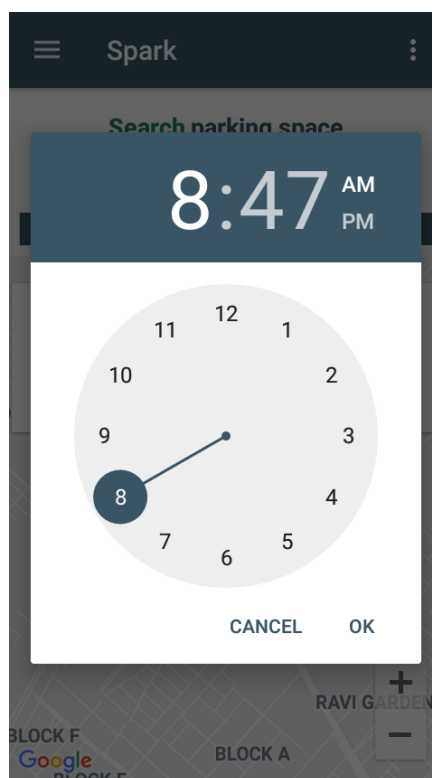


Figure 10-31 Time selection screen

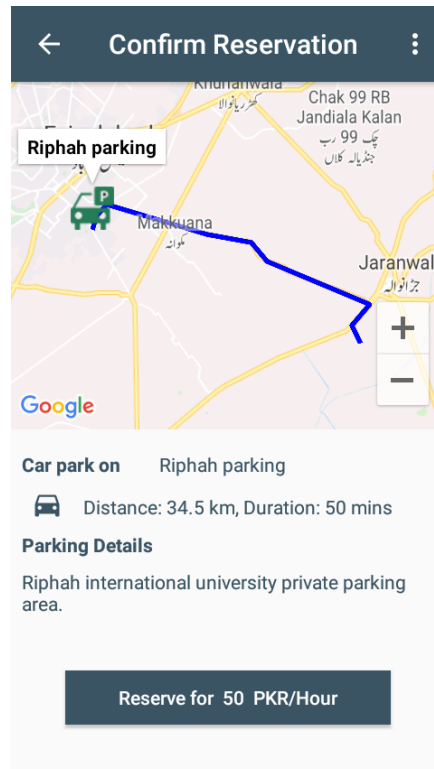


Figure 10-32 Confirm reservation screen

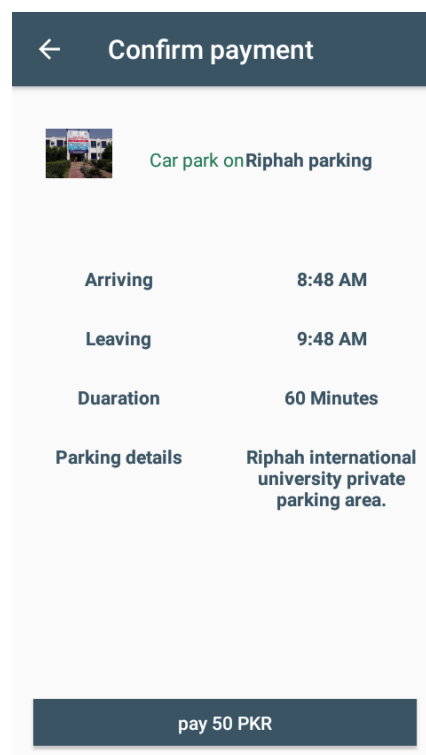


Figure 10-33 Confirm payments screen

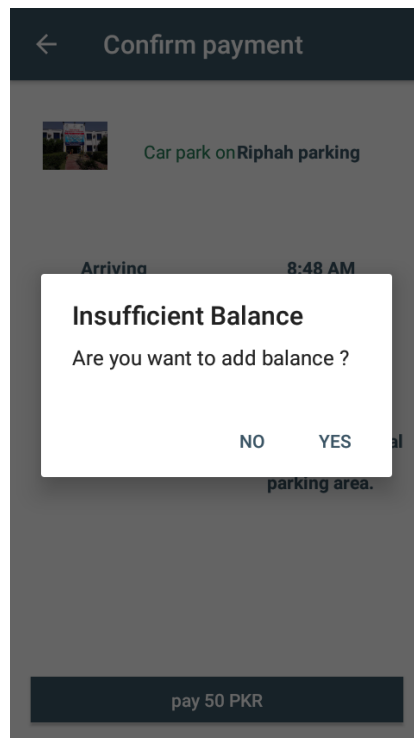


Figure 10-34 Insufficient balance screen

10.1.9 Active session

After success full reservation active session screen will open which contains count down timer to calculate remaining minutes of parking.

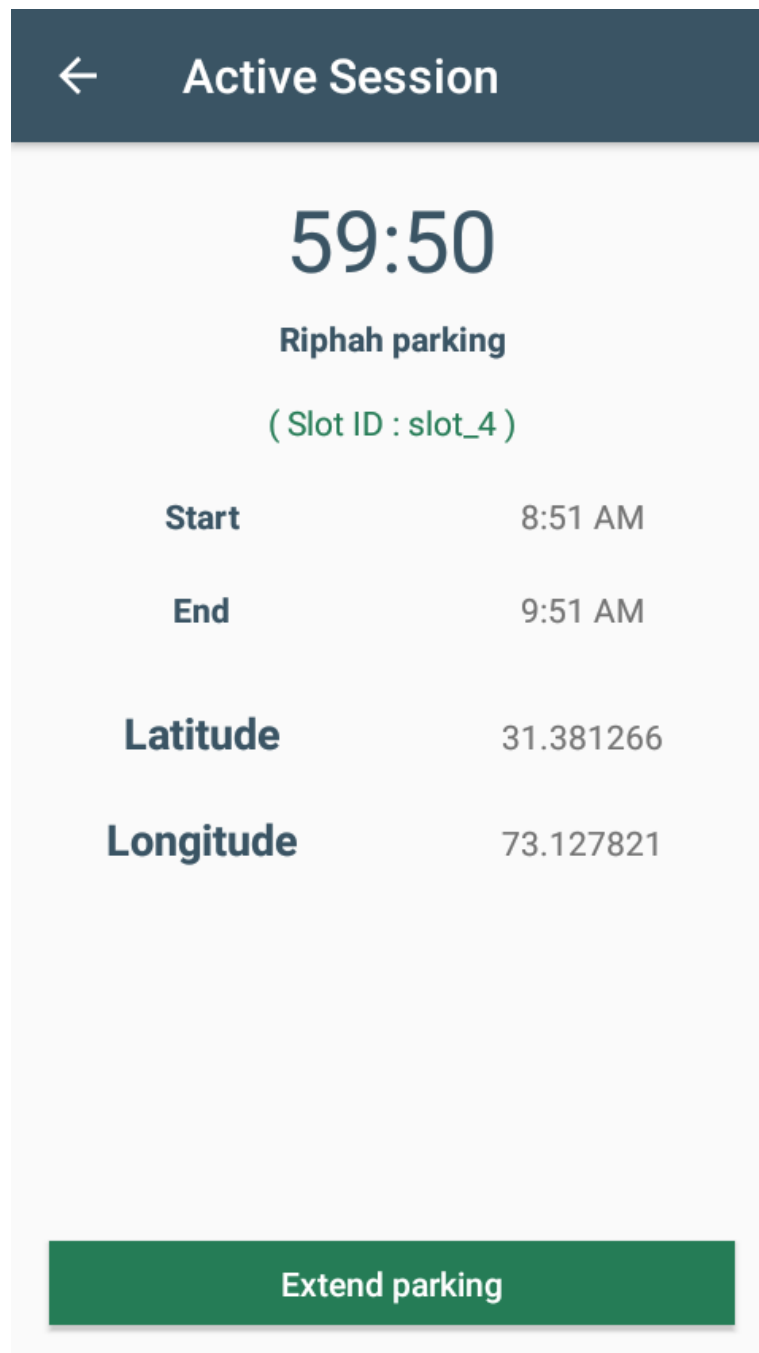


Figure 10-35 Active session screen

10.1.10 Notifications

When your remaining time at parking area will less than five minutes you will be notified, and you will have option to extend parking time. When session expires at parking area you will also notified.

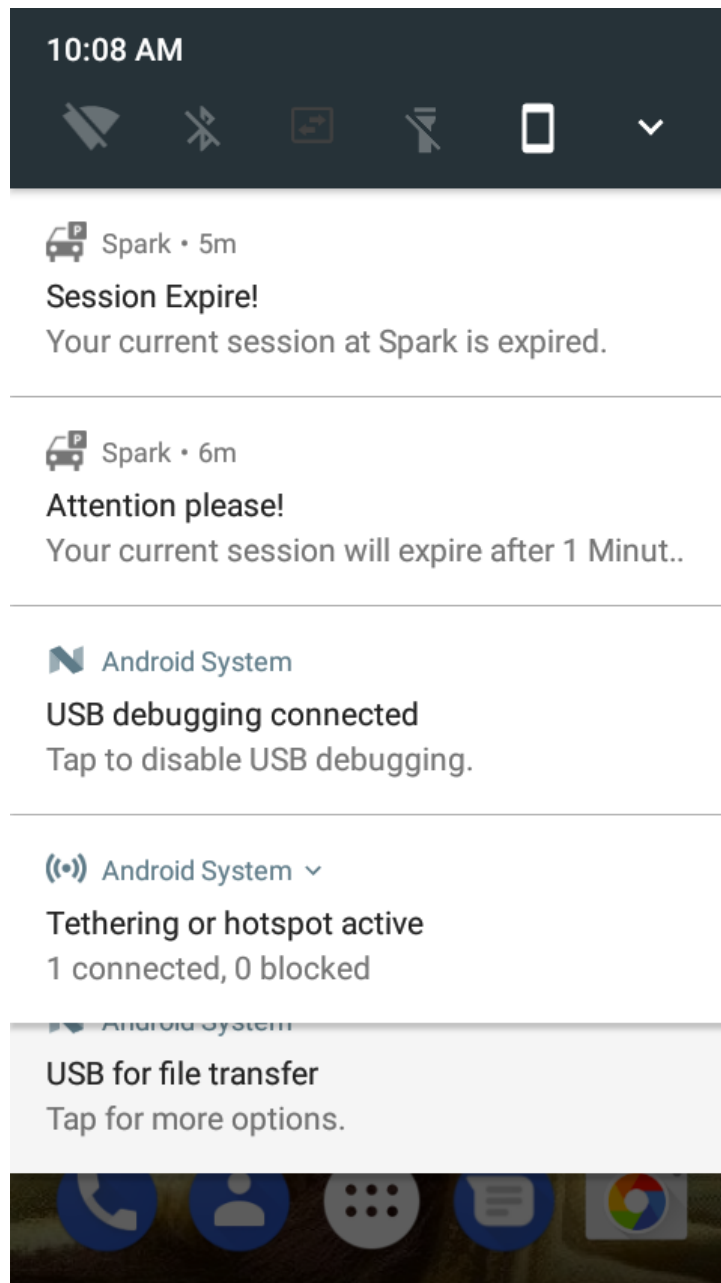


Figure 10-36 Notification screen

10.1.11 Extend parking time

If you want to extend parking, go to main navigation drawer of application then click on active session and follow these instructions.

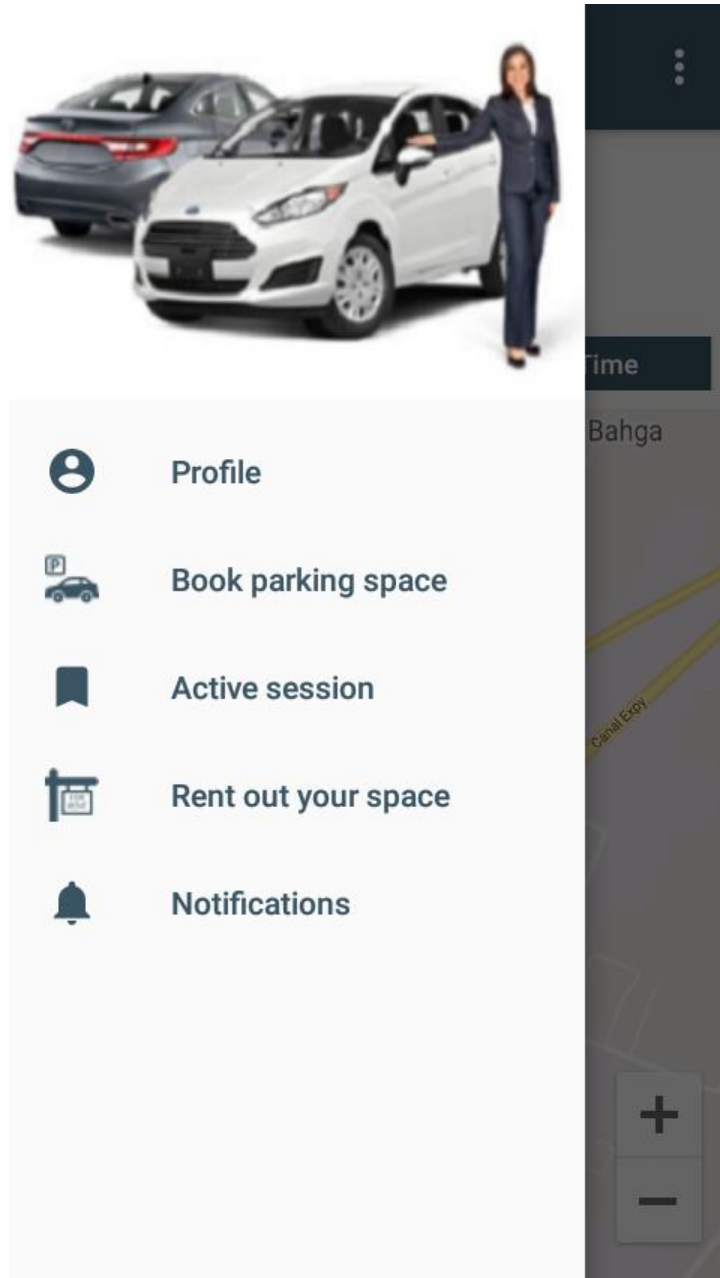


Figure 10-37 Main navigation drawer screen

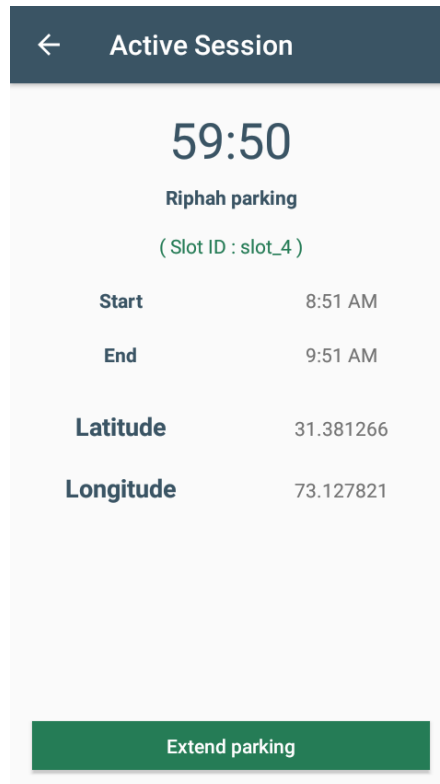


Figure 10-38 Extend parking screen

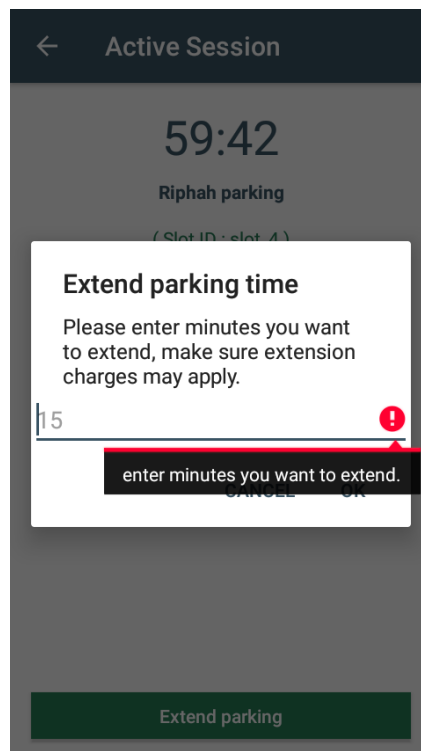


Figure 10-39 Enter minutes to extend screen one

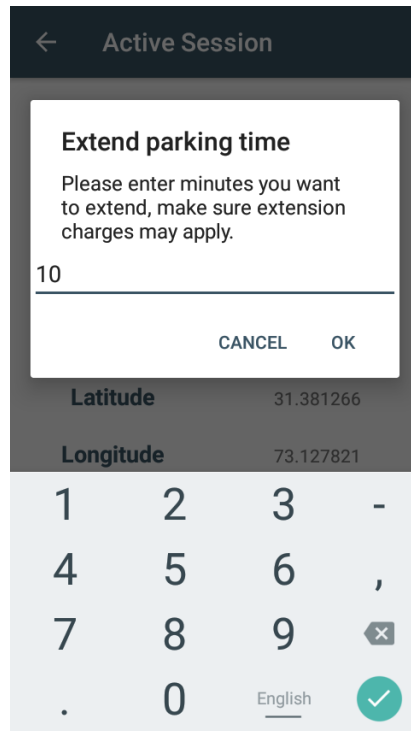


Figure 10-40 Enter minutes to extend screen two

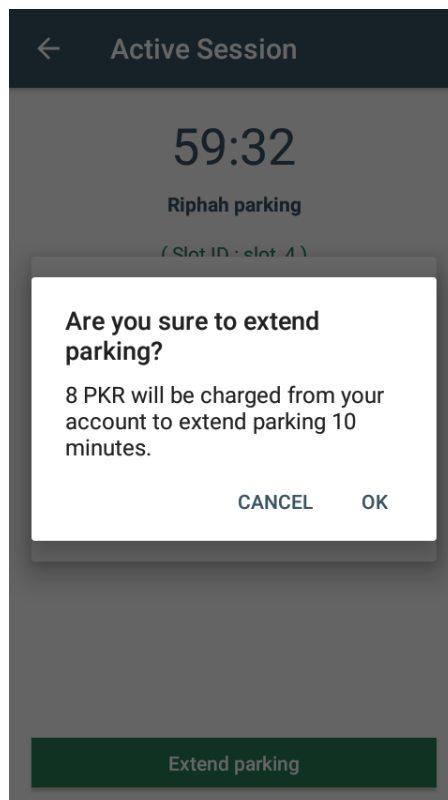


Figure 10-41 Extend parking confirmation screen

10.1.12 Add parking space

Some people have extra space in front of their homes and some organizations have extra parking spaces and they want to use this extra space for public parking. They want to advertise their parking for local business. They can add their extra parking lots in our system and our application user can also reserve parking in their parking area.

← Add parking space

Fill in a few quick details and then sit back to watch the money roll in.

Where is your parking location?

Location

31.286984 73.388736 Find Address

Number of spaces

15

parking title

Testing parking

Parking Details

This is testing parking area

New Parking Space Added.

Add your space

Figure 10-42 Add parking space screen

10.2 Hardware

You have to buy a card from our main office or from parking area this card will provide you a unique id and free balance to book and pay your parking fees. Without this card you cannot park car in parking area. You have to show this card at entrance and system will verify you then you will be allowed to park your car.

Chapter # 11

Bibliography

11 Bibliography

11.1 Appendix A List of Figures

Figure 1-1 SDLC - Iterative Model	5
Figure 2-1 Active Infrared Sensor	12
Figure 2-2 Passive infrared sensor	12
Figure 2-3 Piezoelectric sensor	13
Figure 2-4 Ultrasonic sensor	14
Figure 5-1 Smart parking system use case diagram.....	26
Figure 5-2 Use case diagram for unauthorized user	27
Figure 6-1 Detailed use case diagram for authorized user.....	29
Figure 6-2 Spark application overall flow chart	31
Figure 6-3 Spark hardware overall flow chart	32
Figure 6-4 Sequence diagram for spark system.....	34
Figure 6-5 prototype diagram I	35
Figure 6-6 prototype diagram II.....	35
Figure 6-7 Circuit diagram I	36
Figure 6-8 Circuit diagram II.....	36
Figure 7-1 Android studio IDE	38
Figure 7-2 Firebase	40
Figure 7-3 NodeMCU microcontroller	42
Figure 7-4 HC-SR04 Ultrasonic Sensor	42
Figure 7-5 Servomotor.....	43
Figure 7-6 LCD for display.....	43

Figure 7-7 RFID system.....	44
Figure 7-8 Wi-Fi module	44
Figure 7-9 I2C LCD.....	45
Figure 10-1 Starting screen of Spark application	58
Figure 10-2 First name input screen	59
Figure 10-3 Error message for empty first name	59
Figure 10-4 Second name input screen	60
Figure 10-5 Input screen for email address and password.....	60
Figure 10-6 Filled email and password screen	61
Figure 10-7 Input scree for valid phone number	61
Figure 10-8 Why we need your phone number screen	62
Figure 10-9 Welcome screen of Spark application.....	62
Figure 10-10 Location permission screen.....	63
Figure 10-11 Application information screen.....	63
Figure 10-12 Permissions screen	64
Figure 10-13 Allowed permissions screen.....	64
Figure 10-14 Enable GPS screen	65
Figure 10-15 Location settings screen	65
Figure 10-16 Enabled GPS screen	66
Figure 10-17 Navigation drawer screen.....	66
Figure 10-18 Profile settings screen	67
Figure 10-19 Select image from phone screen	67
Figure 10-20 Crop, rotate, flip profile photo screen	68
Figure 10-21 Uploading profile photo screen.....	68

Figure 10-22 Changed profile photo screen.....	69
Figure 10-23 Edit first name screen.....	69
Figure 10-24 Edit last name screen.....	70
Figure 10-25 Verify email address screen	70
Figure 10-26 Account verification screen.....	71
Figure 10-27 Verified account screen.....	71
Figure 10-28 Search parking space screen.....	72
Figure 10-29 Selected parking area screen	73
Figure 10-30 Select arrival and departure time screen	74
Figure 10-31 Time selection screen.....	74
Figure 10-32 Confirm reservation screen	75
Figure 10-33 Confirm payments screen.....	75
Figure 10-34 Insufficient balance screen	76
Figure 10-35 Active session screen	77
Figure 10-36 Notification screen	78
Figure 10-37 Main navigation drawer screen	79
Figure 10-38 Extend parking screen	80
Figure 10-39 Enter minutes to extend screen one.....	80
Figure 10-40 Enter minutes to extend screen two	81
Figure 10-41 Extend parking confirmation screen	81
Figure 10-42 Add parking space screen.....	82

11.2 Appendix B List of Tables

Table 1 Table of conventions.....	a
-----------------------------------	---

Table 2 Overview of smart parking based on different techniques	10
Table 3 Testing and results	54

Chapter # 12

References

12 References

- [1] A. Lester, Project Management, Planning and Control: Managing Engineering, Construction and Manufacturing Projects to PMI, APM and BSI Standards. Elsevier, 2006.
- [2] “What is fuzzy logic? - Definition from WhatIs.com,” SearchEnterpriseAI. [Online]. Available: <https://searchenterpriseai.techtarget.com/definition/fuzzy-logic>. [Accessed: 12-Aug-2018].
- [3] W. Kaiyu, Z. Yu, S. Guan, X. Yang, M. Sheng, and Z. Tang, “Research and Implementation of Automatic Fuzzy Garage Parking System Based on FPGA,” MATEC Web Conf., vol. 75, p. 07004, Jan. 2016.
- [4] Y. Geng and C. G. Cassandras, “A new ‘Smart Parking’ System Infrastructure and Implementation,” Procedia - Soc. Behav. Sci., vol. 54, pp. 1278–1287, Oct. 2012.
- [5] E. O. Orinda and M. C. Lyamba, “Development of a Global Positioning System (GPS) for Managing Parking,” vol. 1, no. 11, p. 17.
- [6] H. R. H. Al-Absi, J. D. D. Devaraj, P. Sebastian, and Y. V. Voon, “Vision-based automated parking system,” in 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010), 2010, pp. 757–760.
- [7] M. Patil and V. N. Bhonge, “Wireless Sensor Network and RFID for Smart Parking System,” vol. 3, no. 4, p. 5, 2013.
- [8] “Flowchart - Process Flow Charts, Templates, How To, and More.” [Online]. Available: <https://www.smartdraw.com/flowchart/>. [Accessed: 12-Aug-2018].
- [9] “What is Sequence Diagram?” [Online]. Available: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>. [Accessed: 12-Aug-2018].
- [10] “Android (operating system),” p. 18.
- [11] “(1) What is Firebase? - Quora.” [Online]. Available: <https://www.quora.com/What-is-firebase>. [Accessed: 12-Aug-2018].

[12] A. Ashwini, “Using Firebase To Provide Real-time Notifications,” The Startup, 31-Oct-2017. .